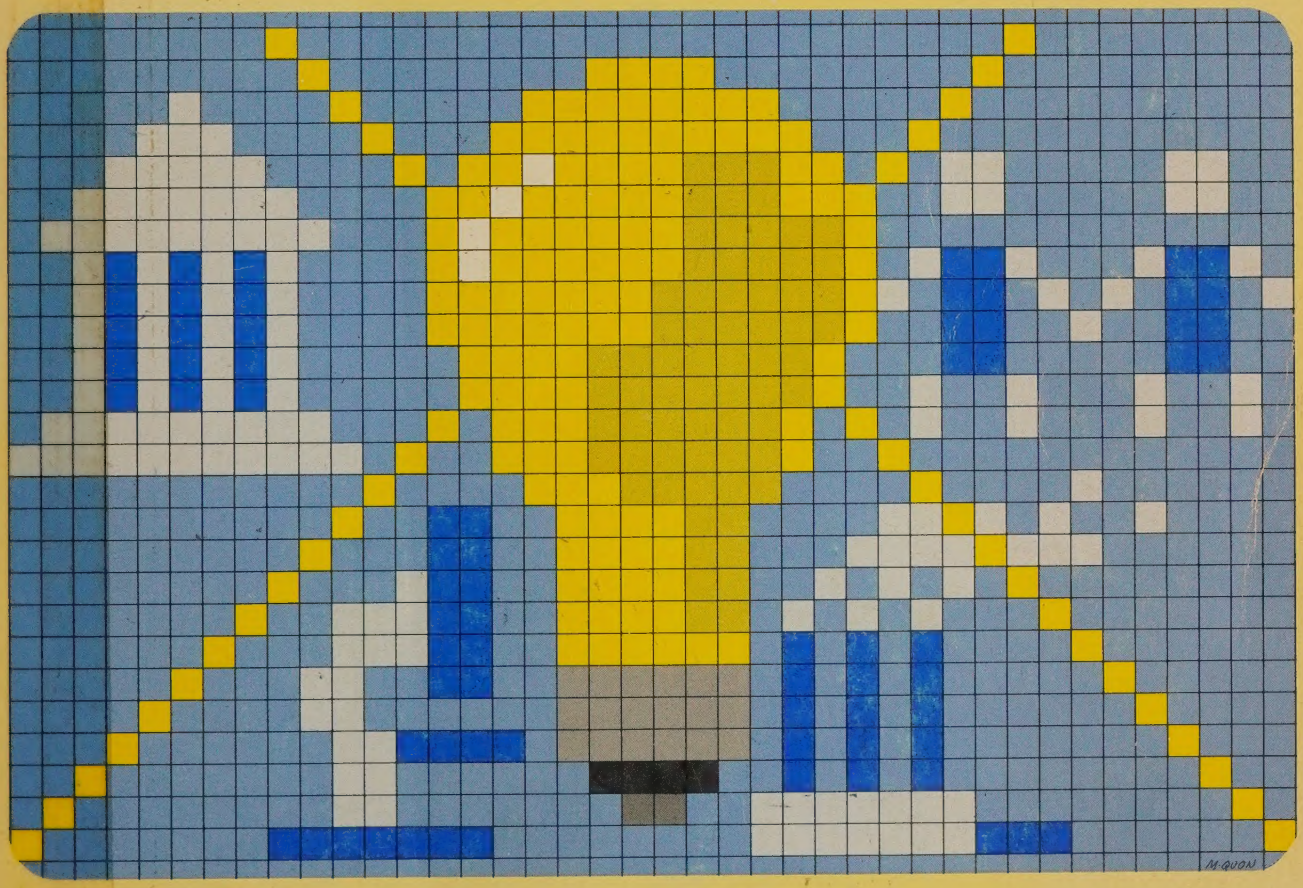
371.39445
C738s

COMPUTERS IN SCIENCE AND SOCIAL STUDIES

A SOURCEBOOK OF IDEAS

Edited By David H. Ahl



49 thought-provoking simulations, applications and educational games for microcomputers! User-tested programs are included.

Digitized by the Internet Archive
in 2024

https://archive.org/details/computersinscien0000davi

Computers in Science and Social Studies:

A Sourcebook of Ideas

159034

Computers in Science and Social Studies:

A Sourcebook of Ideas

Edited by David H. Ahl

371.39445
C738s

Creative Computing Press, Morris Plains, New Jersey

Alverno College
Library Media Center
Milwaukee, Wisconsin

Copyright © 1983 by Creative Computing Press.

All rights reserved. No portion of this book may be reproduced — mechanically, electronically, or by any other means, including photocopying — without written permission of the publisher.

Library of Congress Number: 82-084478
ISBN: 0-916688-44-5

10 9 8 7 6 5 4 3 2 1

Creative Computing Press
39 East Hanover Avenue
Morris Plains, NJ 07950

Dedication

To Camille and Felix

Table Of Contents

Preface viii

Chapter One

Simulations for Social Studies
Hammurabi 2
King 4
Dukedom 6
Computer Simulations in the Classroom 12
Civil War 16
Fur Trader 20
Oregon Trail 22
Lost & Forgotten Island 30
Computers and the Population Problem 35
World Model 36
A Social Science Survey Program 42
Star Merchant 46
Widget 56
Oil Company 57
Electric Company 62
Apple Nuclear Power Plant 66
PET Nuclear Power Plant 74
Geography 81
CMAPS: Chloropleth Mapping 83
Dogbite 86
Map of the 1980's 89

Chapter Two

Ecology and Biology
Seeing is Believing 92
Ecology: A Simulation Model 93
NICHE: A Basic Game of Ecology 98
Odell 2: An Elementary Simulation 103
Swarms 111
Genetic Calculations 117
Genetic Engineering 119
Genetics Problem Solving 123
Genepool 126
Wasps and the Impossible Elevator 132
Guess My Animal 136
Game of Life 138
Life 141
A Faster Life 143

Chapter Three

Health and Psychology
Medical Simulations 146
Smokers' Self-Assessment Test 150
Life Expectancy 154
Building a Mouse 157

Chapter Four

Physics
Projectile Motion 166
Gunner 168
Kinema 169
Lunar 170
Drag 171
Landing Simulator 172
Zeno's Spaceships 177
Planet of the Double Sun 179

Chapter Five

Chemistry
Inorganic Chemistry Program 186
Demonstrating the Gas Laws 188
Periodic Table 191

Preface

Although computers have been with us for over a quarter of a century, most people have strong preconceived notions of what a computer can and can't do. Most frequently it is seen as a tool for solving mathematical problems. Business and financial record keeping and word processing are also seen as appropriate applications for the computer.

But these applications barely scratch the surface of possibilities. The number of new applications for which the computer can be a valuable tool astonishes the imagination. This is good, because as Mary Midgley wrote in *Beast and Man*, "It is an essential feature of vigorous human life that we are constantly being confronted with people who react to the world so differently from ourselves that they stagger us. The whole range of human possibilities is so much wider than any one of us can encompass, that people of the most diverse sorts are needed to cover it. The full human sound goes up only from the whole orchestra."

In addition to being diverse, new applications for the computer, especially in education, ought to be worthwhile and understandable. I am appalled at the number of software packages that are no more than a few pages of a textbook transcribed into a computer program. Equally appalling are the unrealistic problems that textbook authors dream up to illustrate, say, an algebraic principal. Many of these have found their way into educational software packages.

Still other problems may be interesting, but is it worthwhile to try to solve them? Max Black, Professor of Philosophy Emeritus at Cornell, expressed the fear of perhaps every academic researcher — that you could spend your life imagining that you are working on a real problem that isn't one. For example, the "heap dilemma." One grain of corn clearly is not a heap. Two grains are not a heap. Three grains are not a heap. But a pile four feet high is a heap. When did the collection of grains become a heap? Where do you draw the line? Indeed, can you draw such a line?

More important, is the heap dilemma worth solving? Perhaps to a philosopher, but probably not to a ninth grade student. The point is, we can get hung up in all sorts of ethical and philosophical questions, but what really matters is that you are convinced in your own mind that the computer application you want to do is worthwhile and beneficial to mankind.

Educational computer applications should also provide a challenge to the user. As Einstein said, "I have no patience with scientists who take a board, look for its thinnest part, and pound all their nails in there."

The applications in this book were selected to match as closely as possible the criteria stated above. They are pragmatic, classroom-tested applications. They have been proven worthwhile in actual use. Users will be challenged by the applications as they are presented, but even more by thinking about how to improve upon them. And there is enormous diversity.

The majority of applications in this book are simulations. They are not mathematical problems to be solved or drill and practice exercises. Simulations allow users to ask the all-important "what if?" questions. Most simulations have no set answers. They simulate something as close as possible to the real thing and

allow the user to manipulate the variables. By so doing, the user should gain a good understanding of how the particular something works and what can be done to change it.

In the broad area of social studies, the simulations tend to put the user into the role of managing something — for example, managing a family making a tough journey in 1850 over the Oregon Trail. A series of three simulations, Hammurabi, King, and Dukedom, allows the user to manage a self-contained kingdom of increasing complexity.

Several simulations put the user in the role of a manager in a company of some sort, an electric company, oil company, nuclear power plant or trading starship of the future.

Simulations are valuable in that they cut across traditional subject area boundaries. That's the way the real world is; nothing exists in a self-contained box. A decision to control an insect pest, for example, has far-reaching consequences for the economy (cost of damage, cost of control), the ecology (effect on other insects, on birds or animals, on man), employment, food prices and more.

Thus it is with the simulations in this book. While some are labeled "social studies," they could have been labeled "ecology" or "biology" and vice versa.

In general, the ecology/biology section includes applications in two broad areas, simulations of ecological systems and various programs in genetics. But don't be misled by labels or even the title of an application; many have implications far beyond what the title might lead you to suspect.

Rounding out the book are sections on physics and chemistry. Simulations of a lunar lander and artillery cannon, favorites of teachers and students since the 60's are included, as well as recent applications such as calculating planetary orbits and demonstrating the gas laws.

This book puts the emphasis on thinking, not memorizing. Moreover, these applications focus on "learning to learn" rather than mere facts and data. But keep in mind as you discover a new world inside your computer that that world is not an end in itself, but a beginning to understanding our real world and making it a better place in which to live.

Morris Plains, New Jersey
January 1983

David H. Ahl

Chapter One

Simulations for Social Studies

Hammurabi

In this game you direct the administrator of Sumeria, Hammurabi, how to manage the city. The city initially has 1,000 acres, 100 people and 3,000 bushels of grain in storage.

You may buy and sell land with your neighboring city-states for bushels of grain — the price will vary between 17 and 26 bushels per acre. You also must use grain to feed your people and as seed to plant the next year's crop.

You will quickly find that a certain number of people can only tend a certain amount of land and that people starve if they are not fed enough. You also have the unexpected to contend with such as a plague, rats destroying stored grain, and variable harvests.

You will also find that managing just the few resources in this game is not a trivial job over a period of say ten years. The crisis of population density rears its head very rapidly.

This program was originally written in Focal at DEC; author unknown. David Ahl converted it to BASIC and added the 10-year performance accessment. If you wish to change any of the factors, the extensive remarks in the program should make modification fairly straightforward.

Note for trivia buffs: somewhere along the line an m was dropped out of the spelling of Hammurabi in the Ahl version of the computer program. This error has spread far and wide until a generation of students who have used this program now think that Hammurabi is the incorrect spelling.

```
                HAMURABI
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

TRY YOUR HAND AT GOVERNING ANCIENT SUMERIA
FOR A TEN-YEAR TERM OF OFFICE.

HAMURABI:  I BEG TO REPORT TO YOU,
IN YEAR 1 , 0 PEOPLE STARVED, 5 CAME TO THE CITY,
POPULATION IS NOW 100
THE CITY NOW OWNS 1000 ACRES.
YOU HARVESTED 3 BUSHELS PER ACRE.
RATS ATE 200 BUSHELS.
YOU NOW HAVE 2800 BUSHELS IN STORE.

LAND IS TRADING AT 24 BUSHELS PER ACRE.
HOW MANY ACRES DO YOU WISH TO BUY? 10

HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE? 2000

HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED? 990

HAMURABI:  I BEG TO REPORT TO YOU,
IN YEAR 2 , 0 PEOPLE STARVED, 5 CAME TO THE CITY,
POPULATION IS NOW 105
THE CITY NOW OWNS 1010 ACRES.
YOU HARVESTED 3 BUSHELS PER ACRE.
RATS ATE 16 BUSHELS.
YOU NOW HAVE 3019 BUSHELS IN STORE.

LAND IS TRADING AT 21 BUSHELS PER ACRE.
HOW MANY ACRES DO YOU WISH TO BUY? 25

HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE? 2000

HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED? 1000
HAMURABI:  THINK AGAIN.  YOU HAVE ONLY
 494 BUSHELS OF GRAIN.  NOW THEN,
HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED? 500

HAMURABI:  I BEG TO REPORT TO YOU,
IN YEAR 3 , 5 PEOPLE STARVED, 5 CAME TO THE CITY,
A HORRIBLE PLAGUE STRUCK!  HALF THE PEOPLE DIED.
POPULATION IS NOW 52
THE CITY NOW OWNS 1035 ACRES.
YOU HARVESTED 1 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YOU NOW HAVE 744 BUSHELS IN STORE.

LAND IS TRADING AT 17 BUSHELS PER ACRE.
HOW MANY ACRES DO YOU WISH TO BUY? 0
HOW MANY ACRES DO YOU WISH TO SELL? 25

HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE? 1000

HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED? 500
HAMURABI:  THINK AGAIN.  YOU HAVE ONLY
 169 BUSHELS OF GRAIN.  NOW THEN,
HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED? 300


HAMURABI:  I BEG TO REPORT TO YOU,
IN YEAR 4 , 2 PEOPLE STARVED, 12 CAME TO THE CITY,
POPULATION IS NOW 62
THE CITY NOW OWNS 1010 ACRES.
YOU HARVESTED 1 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YOU NOW HAVE 319 BUSHELS IN STORE.

LAND IS TRADING AT 23 BUSHELS PER ACRE.
HOW MANY ACRES DO YOU WISH TO BUY? 0
HOW MANY ACRES DO YOU WISH TO SELL? 500

HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE? 500

HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED? 230

YOU STARVED 37 PEOPLE IN ONE YEAR!!!
DUE TO THIS EXTREME MISMANAGEMENT YOU HAVE NOT ONLY
BEEN IMPEACHED AND THROWN OUT OF OFFICE BUT YOU HAVE
ALSO BEEN DECLARED NATIONAL FINK!!!!

SO LONG FOR NOW.
```

```
10 PRINT TAB(32);"HAMURABI"
20 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
30 PRINT:PRINT:PRINT
80 PRINT "TRY YOUR HAND AT GOVERNING ANCIENT SUMERIA"
90 PRINT "FOR A TEN-YEAR TERM OF OFFICE.":PRINT
95 D1=0: P1=0
100 Z=0: P=95:S=2800: H=3000: E=H-S
110 Y=3: A=H/Y: I=5: Q=1
210 D=0
215 PRINT:PRINT:PRINT "HAMURABI:  I BEG TO REPORT TO YOU,": Z=Z+1
217 PRINT "IN YEAR";Z;",";D;"PEOPLE STARVED,";I;"CAME TO THE CITY,"
218 P=P+I
227 IF Q>0 THEN 230
228 P=INT(P/2)
229 PRINT "A HORRIBLE PLAGUE STRUCK!  HALF THE PEOPLE DIED."
230 PRINT "POPULATION IS NOW";P
232 PRINT "THE CITY NOW OWNS";A;"ACRES."
235 PRINT "YOU HARVESTED";Y;"BUSHELS PER ACRE."
250 PRINT "RATS ATE";E;"BUSHELS."
260 PRINT "YOU NOW HAVE";S;"BUSHELS IN STORE.": PRINT
270 IF Z=11 THEN 860
310 C=INT(10*RND(1)): Y=C+17
312 PRINT "LAND IS TRADING AT";Y;"BUSHELS PER ACRE."
320 PRINT "HOW MANY ACRES DO YOU WISH TO BUY";
321 INPUT Q: IF Q<0 THEN 850
322 IF Y*Q<=S THEN 330
323 GOSUB 710
324 GOTO 320
330 IF Q=0 THEN 340
331 A=A+Q: S=S-Y*Q: C=0
334 GOTO 400
340 PRINT "HOW MANY ACRES DO YOU WISH TO SELL";
341 INPUT Q: IF Q<0 THEN 850
342 IF Q<A THEN 350
343 GOSUB 720
344 GOTO 340
350 A=A-Q: S=S+Y*Q: C=0
400 PRINT
410 PRINT "HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE";
411 INPUT Q
412 IF Q<0 THEN 850
418 REM *** TRYING TO USE MORE GRAIN THAN IS IN SILOS?
420 IF Q<=S THEN 430
421 GOSUB 710
422 GOTO 410
430 S=S-Q: C=1: PRINT
440 PRINT "HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED";
441 INPUT D: IF D=0 THEN 511
442 IF D<0 THEN 850
444 REM *** TRYING TO PLANT MORE ACRES THAN YOU OWN?
445 IF D<=A THEN 450
446 GOSUB 720
447 GOTO 440
449 REM *** ENOUGH GRAIN FOR SEED?
450 IF INT(D/2)<=S THEN 455
452 GOSUB 710
453 GOTO 440
454 REM *** ENOUGH PEOPLE TO TEND THE CROPS?
455 IF D<10*P THEN 510
460 PRINT "BUT YOU HAVE ONLY";P;"PEOPLE TO TEND THE FIELDS!  NOW THEN,"
470 GOTO 440
510 S=S-INT(D/2)
511 GOSUB 800
512 REM *** A BOUNTIFUL HARVEST!
515 Y=C: H=D*Y: E=0
521 GOSUB 800
522 IF INT(C/2)<>C/2 THEN 530
523 REM *** RATS ARE RUNNING WILD!!
525 E=INT(S/C)
530 S=S-E+H
531 GOSUB 800
532 REM *** LET'S HAVE SOME BABIES
533 I=INT(C*(20*A+S)/P/100+1)
539 REM *** HOW MANY PEOPLE HAD FULL TUMMIES?
540 C=INT(Q/20)
541 REM *** HORROR, A 15% CHANCE OF PLAGUE
542 Q=INT(10*(2*RND(1)-.3))
550 IF P<C THEN 210
551 REM *** STARVE ENOUGH FOR IMPEACHMENT?
552 D=P-C: IF D>.45*P THEN 560
553 P1=((Z-1)*P1+D*100/P)/Z
555 P=C: D1=D1+D: GOTO 215
560 PRINT: PRINT "YOU STARVED";D;"PEOPLE IN ONE YEAR!!!"
565 PRINT "DUE TO THIS EXTREME MISMANAGEMENT YOU HAVE NOT ONLY"
566 PRINT "BEEN IMPEACHED AND THROWN OUT OF OFFICE BUT YOU HAVE"
567 PRINT "ALSO BEEN DECLARED NATIONAL FINK!!!!": GOTO 990
710 PRINT "HAMURABI:  THINK AGAIN.  YOU HAVE ONLY"
711 PRINT S;"BUSHELS OF GRAIN.  NOW THEN,"
712 RETURN
720 PRINT "HAMURABI:  THINK AGAIN.  YOU OWN ONLY";A;"ACRES.  NOW THEN,"
730 RETURN
800 C=INT(RND(1)*5)+1
801 RETURN
850 PRINT: PRINT "HAMURABI:  I CANNOT DO WHAT YOU WISH."
855 PRINT "GET YOURSELF ANOTHER STEWARD!!!!!"
857 GOTO 990
860 PRINT "IN YOUR 10-YEAR TERM OF OFFICE,";P1;"PERCENT OF THE"
862 PRINT "POPULATION STARVED PER YEAR ON THE AVERAGE, I.E. A TOTAL OF"
865 PRINT D1;"PEOPLE DIED!!": L=A/P
870 PRINT "YOU STARTED WITH 10 ACRES PER PERSON AND ENDED WITH"
875 PRINT L;"ACRES PER PERSON.": PRINT
880 IF P1>33 THEN 565
885 IF L<7 THEN 565
890 IF P1>10 THEN 940
892 IF L<9 THEN 940
895 IF IF P1>3 THEN 960
896 IF L<10 THEN 960
900 PRINT "A FANTASTIC PERFORMANCE!!!  CHARLEMAGNE, DISRAELI, AND"
905 PRINT "JEFFERSON COMBINED COULD NOT HAVE DONE BETTER!":GOTO 990
940 PRINT "YOUR HEAVY-HANDED PERFORMANCE SMACKS OF NERO AND IVAN IV."
945 PRINT "THE PEOPLE (REMAINING) FIND YOU AN UNPLEASANT RULER, AND,"
950 PRINT "FRANKLY, HATE YOUR GUTS!!":GOTO 990
960 PRINT "YOUR PERFORMANCE COULD HAVE BEEN SOMEWHAT BETTER, BUT"
965 PRINT "REALLY WASN'T TOO BAD AT ALL. ";INT(P*.8*RND(1));"PEOPLE"
970 PRINT "DEARLY LIKE TO SEE YOU ASSASSINATED BUT WE ALL HAVE OUR"
975 PRINT "TRIVIAL PROBLEMS."
990 PRINT: FOR N=1 TO 10: PRINT CHR$(7);: NEXT N
995 PRINT "SO LONG FOR NOW.": PRINT
999 END
```

King

This is one of the more comprehensive, difficult, and interesting land and resource management games. (If you've never played one of these games, start with HAMMURABI.)

In this game, you are Premier of Setats Detinu, a small communist island 30 by 70 miles long. Your job is to decide upon the budget of the country and distribute money to your countrymen from the communal treasury.

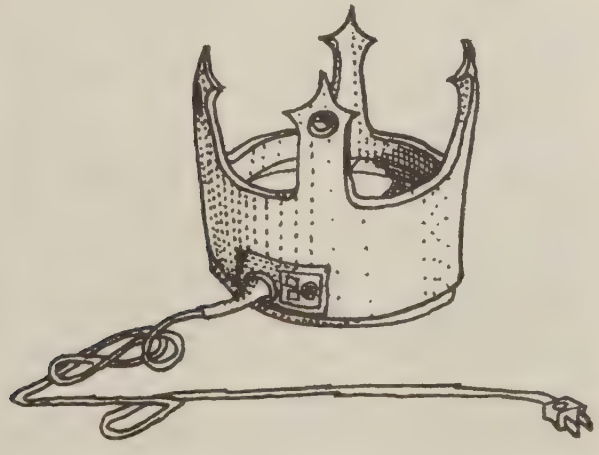
The money system is Rallods; each person needs 100 Rallods per year to survive. Your country's income comes from farm produce and tourists visiting your magnificent forests, hunting, fishing, etc. Part of your land is farm land but it also has an excellent mineral content and may be sold to foreign industry for strip mining. Industry import and support their own workers. Crops cost between 10 and 15 Rallods per square mile to plant, cultivate, and harvest. Your goal is to complete an eight-year term of office without major mishap. A word of warning: it isn't easy!

The author of this program is James A. Storer who wrote it while a student at Lexington High School.

```
                KING
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


DO YOU WANT INSTRUCTIONS? YES


CONGRATULATIONS! YOU'VE JUST BEEN ELECTED PREMIER OF SETATS
DETINU, A SMALL COMMUNIST ISLAND 30 BY 70 MILES LONG.  YOUR
JOB IS TO DECIDE UPON THE COUNTRY'S BUDGET AND DISTRIBUTE
MONEY TO YOUR COUNTRYMEN FROM THE COMMUNAL TREASURY.
THE MONEY SYSTEM IS RALLODS, AND EACH PERSON NEEDS 100
RALLODS PER YEAR TO SURVIVE.  YOUR COUNTRY'S INCOME COMES
FROM FARM PRODUCE AND TOURISTS VISITING YOUR MAGNIFICENT
FORESTS, HUNTING, FISHING, ETC.  HALF YOUR LAND IS FARM LAND
WHICH ALSO HAS AN EXCELLENT MINERAL CONTENT AND MAY BE SOLD
TO FOREIGN INDUSTRY (STRIP MINING) WHO IMPORT AND SUPPORT
THEIR OWN WORKERS.  CROPS COST BETWEEN 10 AND 15 RALLODS PER
SQ. MILE TO PLANT.
YOUR GOAL IS TO COMPLETE YOUR 8 YEAR TERM OF OFFICE.
GOOD LUCK.
```

```
YOU NOW HAVE 59907 RALLODS IN THE TREASURY.
 502 COUNTRYMEN, AND 2000 SQ. MILES OF LAND.
THIS YEAR INDUSTRY WILL BUY LAND FOR 103 RALLODS PER SQ. MILE.
LAND CURRENTLY COSTS 10 RALLODS PER SQ. MILE TO PLANT.


HOW MANY SQ. MILES DO YOU WISH TO SELL TO INDUSTRY? 200
HOW MANY RALLODS WILL YOU DISTRIBUTE TO YOUR COUNTRYMEN? 50200
HOW MANY SQ. MILES DO YOU WISH TO PLANT? 500
HOW MANY RALLODS DO YOU WISH TO SPEND ON POLLUTION CONTROL? 10000


 212 WORKERS CAME TO THE COUNTRY AND 396 COUNTRYMEN CAME TO THE ISLAND.
 OF 500 SQ. MILES PLANTED, YOU HARVESTED 340 SQ. MILES OF CROPS.
   (DUE TO AIR AND WATER POLLUTION FROM FOREIGN INDUSTRY.)
 MAKING 17510 RALLODS.
 YOU MADE 8179 RALLODS FROM TOURIST TRADE.

YOU NOW HAVE 65634 RALLODS IN THE TREASURY.
 898 COUNTRYMEN,  212 FOREIGN WORKERS, AND 1800 SQ. MILES OF LAND.
THIS YEAR INDUSTRY WILL BUY LAND FOR 98 RALLODS PER SQ. MILE.
LAND CURRENTLY COSTS 13 RALLODS PER SQ. MILE TO PLANT.

HOW MANY SQ. MILES DO YOU WISH TO SELL TO INDUSTRY? 0
HOW MANY RALLODS WILL YOU DISTRIBUTE TO YOUR COUNTRYMEN? 89800
   THINK AGAIN, YOU'VE ONLY 65634 RALLODS IN THE TREASURY
HOW MANY RALLODS WILL YOU DISTRIBUTE TO YOUR COUNTRYMEN? 40000
HOW MANY SQ. MILES DO YOU WISH TO PLANT? 600
HOW MANY RALLODS DO YOU WISH TO SPEND ON POLLUTION CONTROL? 6000


 498 COUNTRYMEN DIED OF STARVATION
   YOU WERE FORCED TO SPEND 4482 RALLODS ON FUNERAL EXPENSES
 236 COUNTRYMEN CAME TO THE ISLAND.
 OF 600 SQ. MILES PLANTED, YOU HARVESTED 448 SQ. MILES OF CROPS.
   (DUE TO AIR AND WATER POLLUTION FROM FOREIGN INDUSTRY.)
 MAKING 21952 RALLODS.
 YOU MADE 6068 RALLODS FROM TOURIST TRADE.
   DECREASE BECAUSE AIR POLLUTION IS KILLING GAME BIRD POPULATION.


 498 COUNTRYMEN DIED IN ONE YEAR!!!!!
DUE TO THIS EXTREME MISMANAGEMENT YOU HAVE NOT ONLY
BEEN IMPEACHED AND THROWN OUT OF OFFICE BUT YOU
HAVE ALSO GAINED A VERY BAD REPUTATION.
```

```
1 PRINT TAB(34);"KING"
2 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
3 PRINT:PRINT:PRINT
4 PRINT "DO YOU WANT INSTRUCTIONS";
5 INPUT Z$
6 N5=8
10 IF LEFT$(Z$,1)="N" THEN 47
11 IF Z$="AGAIN" THEN 1960
12 PRINT:PRINT:PRINT
20 PRINT "CONGRATULATIONS! YOU'VE JUST BEEN ELECTED PREMIER OF SETATS"
22 PRINT "DETINU, A SMALL COMMUNIST ISLAND 30 BY 70 MILES LONG.  YOUR"
24 PRINT "JOB IS TO DECIDE UPON THE COUNTRY'S BUDGET AND DISTRIBUTE"
26 PRINT "MONEY TO YOUR COUNTRYMEN FROM THE COMMUNAL TREASURY."
28 PRINT "THE MONEY SYSTEM IS RALLODS, AND EACH PERSON NEEDS 100"
30 PRINT "RALLODS PER YEAR TO SURVIVE.  YOUR COUNTRY'S INCOME COMES"
32 PRINT "FROM FARM PRODUCE AND TOURISTS VISITING YOUR MAGNIFICENT"
34 PRINT "FORESTS, HUNTING, FISHING, ETC.  HALF YOUR LAND IS FARM LAND"
36 PRINT "WHICH ALSO HAS AN EXCELLENT MINERAL CONTENT AND MAY BE SOLD"
38 PRINT "TO FOREIGN INDUSTRY (STRIP MINING) WHO IMPORT AND SUPPORT"
40 PRINT "THEIR OWN WORKERS.  CROPS COST BETWEEN 10 AND 15 RALLODS PER"
42 PRINT "SQ. MILE TO PLANT."
44 PRINT "YOUR GOAL IS TO COMPLETE YOUR";N5;"YEAR TERM OF OFFICE."
46 PRINT "GOOD LUCK."
47 PRINT
50 A=INT(60000+(1000*RND(1))-(1000*RND(1)))
55 B=INT(500+(10*RND(1))-(10*RND(1)))
65 D=2000
100 W=INT(10*RND(1)+95)
102 PRINT
105 PRINT "YOU NOW HAVE";A;"RALLODS IN THE TREASURY."
110 PRINT INT(B);"COUNTRYMEN, ";
115 V9=INT(((RND(1)/2)*10+10))
120 IF C=0 THEN 140
130 PRINT INT(C);"FOREIGN WORKERS, ";
140 PRINT "AND";INT(D);"SQ. MILES OF LAND."
150 PRINT "THIS YEAR INDUSTRY WILL BUY LAND FOR";W;
152 PRINT "RALLODS PER SQ. MILE."
155 PRINT "LAND CURRENTLY COSTS";V9;"RALLODS PER SQ. MILE TO PLANT."
162 PRINT
200 PRINT "HOW MANY SQ. MILES DO YOU WISH TO SELL TO INDUSTRY";
210 INPUT H
```



```
215 IF H<0 THEN 200
220 IF H<=D-1000 THEN 300
230 PRINT "   THINK AGAIN, YOU'VE ONLY";D-1000;"SQ. MILES OF FARM LAND"
240 IF X<>0 THEN 200
250 PRINT "(FOREIGN INDUSTRY WILL ONLY BUY FARM LAND BECAUSE"
260 PRINT "FOREST LAND IS UNECONOMICAL TO STRIP MINE DUE TO TREES,"
270 PRINT "THICKER TOP SOIL, ETC.)"
280 X=1
299 GOTO 200
300 D=INT(D-H)
310 A=INT(A+(H*W))
320 PRINT "HOW MANY RALLODS WILL YOU DISTRIBUTE TO YOUR COUNTRYMEN";
340 INPUT I
342 IF I<0 THEN 320
350 IF I<A THEN 400
360 IF I=A THEN 380
370 PRINT "   THINK AGAIN, YOU'VE ONLY";A;"RALLODS IN THE TREASURY"
375 GOTO 320
380 J=0
390 K=0
395 A=0
399 GOTO 1000
400 A=INT(A-I)
410 PRINT "HOW MANY SQ. MILES DO YOU WISH TO PLANT";
420 INPUT J
421 IF J<0 THEN 410
422 IF J<=B*2 THEN 426
423 PRINT "   SORRY, BUT EACH COUNTRYMAN CAN ONLY PLANT 2 SQ. MILES"
424 GOTO 410
426 IF J<=D-1000 THEN 430
427 PRINT "   SORRY, BUT YOU'VE ONLY";D-1000;"SQ. MILES OF FARM LAND"
428 GOTO 410
430 U1=INT(J*V9)
435 IF U1<A THEN 500
440 IF U1=A THEN 490
450 PRINT"   THINK AGAIN, YOU'VE ONLY";A;"RALLODS LEFT IN THE TREASURY"
460 GOTO 410
490 K=0
495 A=0
499 GOTO 1000
500 A=A-U1
510 PRINT "HOW MANY RALLODS DO YOU WISH TO SPEND ON POLLUTION CONTROL";
520 INPUT K
522 IF K<0 THEN 510
530 IF K<=A THEN 1000
540 PRINT "   THINK AGAIN, YOU'VE ONLY";A;"RALLODS REMAINING"
550 GOTO 510
600 IF H<>0 THEN 1002
602 IF I<>0 THEN 1002
604 IF J<>0 THEN 1002
606 IF K<>0 THEN 1002
609 PRINT
612 PRINT "GOODBYE."
614 PRINT "(IF YOU WISH TO CONTINUE THIS GAME AT A LATE DATE, ANSWER"
616 PRINT "'AGAIN' WHEN ASKED IF YOU WANT INSTRUCTIONS, AT THE"
618 STOP
1000 GOTO 600
1002 PRINT
1003 PRINT
1010 A=INT(A-K)
1020 A4=A
1100 IF INT(I/100-B)>=0 THEN 1120
1105 IF I/100<50 THEN 1700
1110 PRINT INT(B-(I/100));"COUNTRYMEN DIED OF STARVATION"
1120 F1=INT(RND(1)*(2000-D))
1122 IF K<25 THEN 1130
1125 F1=INT(F1/(K/25))
```

Dukedom *Richard Kaapke*

DUKEDOM is a much extended version of the popular Hammurabi land management game. The game is fun to play, but it is also an outstanding educational tool for learning to manage limited resources. There are also hazards such as rats, locusts, plagues, taxes, and wars with which to contend.

The version of DUKEDOM presented here runs in a 48K CP/M system with Microsoft Disk BASIC. About 13K of BASIC workspace is used. Thus, with the appropriate conversions, this game will probably run in a 16K Level II TRS-80. It can also be converted to run in Applesoft but will probably take more than 16K because of the more extensive conversions involved.

For Level II TRS-80s: the only known necessary conversions are for LINE INPUT and DEF FN. For the former, use regular INPUT, which works fine although it's not as tidy. To convert DEF FN, expand out the functions in line with the program. For example, line 1050 would become:

```
Q3% = RND(1) * (1 + Q2% - Q1%) + Q1%
```

and line 6500 would become:

```
X1 = (RND(1) * (1 + F3% - (-F3%)) + (-F3%)
     + R(8)) + 4
```

[think about it!]

For users of other computers, we suggest you buy a Level II manual at your local Radio Shack store to find out how Microsoft Extended BASIC works. Disk capabilities are not required except for printing the instruction file, reproduced here verbatim. Enjoy! —SN

```
TYPE B:DUKEDOM.INS

You are one of several Dukes chosen by the High King to help run the
Kingdom.  Your Duchy is not in the best of shape, and your job is to
build up its population, land holdings and grain reserves.  Your secret
ambition is to become powerful enough to overthrow the High King.

The game cycles on an annual basis, and it is now fall and the harvest
has just been completed.  Each year at this time the computer will
display the current population, land and grain totals, followed by a
detailed report of the previous year's events.  Note that land and
grain are measured in metric units hectares (HA.) and hectoliters
(HL.), respectively.

Each year you will have to make the following decisions:

        Grain for food =

You must decide how much grain to feed the peasants.  14 HL. of grain
will just adequately feed one peasant; 13 will cause some hunger and
decrease the peasants' fighting ability, and 12 or fewer will cause
some starvations.  The peasants will complain if you try to starve them
excessively and they know that you are holding back grain.  If you feed
the peasants more than 14 HL. each (up to a maximum of 18) they will
appreciate the boon and fight better in any war the following summer.
A long term memory keeps track of the peasants' cumulative attitude (it
fades slowly with time) and if you create sufficient bad will (by
underfeeding them, for instance) they will depose you.  You may enter
the quantity of grain for the peasants in two ways:  Numbers less than
100 are interpreted as hectoliters-per-peasant, while an entry of 100
or more represents the total amount for the entire population.

        Land to buy at nn HL./HA. =

Enter the number of hectares of land you want to buy.  The prices
offered vary from about 4 hectoliters/hectare to about 30, depending
primarily on last year's crop yield.  If you don't want to buy any
land, enter 0.  You will then be given the option of selling your land
at a price one unit lower than the buying price.  Enter the number of
hectares you want to sell, or enter 0 if you don't want to sell any.

        Land to plant =

Enter the number of hectares you wish to plant.  Each hectare planted
will require 2 hectoliters of grain to seed it.  Also, remember that
each peasant can plant and care for no more than 4 hectares.  There is
no fertilizer and no alternate crop, so land used many years in a row
becomes depleted.  The annual report lists the number of hectares you
have of each of six classes from 100% yield to 0%.  In any given year,
land used in any class moves 1 step closer to being totally depleted
while unused land moves two steps closer to fallow (100%).  The best
quality land will always be planted first.  The yield for fallow land
is calculated each year at random (variances in the weather) and ranges
from 4 to 13 hectoliters of grain harvested for each hectare planted.
The actual yield obtained will be the average generated by the various
qualities of land used.

        Special Operating Instructions

When a response is prompted by a "?", a Y or N may be given for Yes or
No, respectively.  A simple return will be assumed to be a "N" response.

When a response is prompted by an "=", a non-negative integer is
required.  Any fraction will be trimmed from input, and a simple return
will be interpreted as an entry of 0.

This version of DUKEDOM allows the player to have the printing of
the long detailed reports omitted from the game.  A beginner should
respond "N" in order to receive the long but helpful reports.  An
experienced DUKEDOM player seldom if ever needs these reports and
once a response is given, the option is set and the game must be
re-run from the beginning in order to change the option, unless
^C breaking the program and setting R$="Y" or R$="N" (the option
yes/no flag you choose) then CONTinuing is convenient.

*EOF*

Additional information on DUKEDOM (Microcomputer Version)
        or
Welcome to medievalworld!  Boy, have we got a vacation for you...
```

General Information

Running totals are maintained by the computer. All additions and subtractions are made at once and further transactions are limited by the current balance. No credit is allowed (with one exception).

One hectare of land equals about 2.5 acres. One hectoliter of grain equals about 2.8 bushels.

It is (usually) necessary to gamble occasionly to win. Most gambles consist of buying land you can't afford at very low prices and gambling that yield will be high and there won't be a war. If the gamble fails, you will spend the next ten years recovering (if you survive, that is).

Food Allocation

By overfeeding the peasants when possible, you can build up good will among the population. This may save your life as it can counteract unavoidable resentment in the future (during times of famine, for instance). Judge Lynch never sleeps!

Land Trading

When you buy land you always receive 60% quality. When you sell land the machine sells your 60% land until used up, then the 80% quality and finally the 100% if you sell that much. You can never sell 40% (or poorer) quality land; no buyers will accept it.

There is another limit on land sales: You cannot sell more than 4000 HL. worth in any one year. That's all the grain available to pay you with.

Crop Hazards

Sometimes the rats get into the granery and eat up to 10% or so of your reserve grain. Rats never eat field grain -- field grain is eaten by the seven year locusts. They eat half of all your crop in the years that they appear. The yield printed in these years already includes locust losses.

The King's Peasant Levy

Occasionly rats will eat so much of the High King's grain that some of his workers starve to death. When this happens, the King will require some peasants from each of his Dukes as replacements. You may supply them as requested or pay an alternate amount of grain.

Wars

Neighboring Dukes may attack you, hoping to obtain some land. This is more probable in years of poor crop yield. It is no secret and you can attack first if you wish. This means that you and your peasants go over there some night and burn a few huts and generally make a great din. If your attack is impressive, the nearby Duke may cancel his war plans. This depends on the size of your attack force and the size of his current defense force. You will certainly lose some peasants in such an attack.

If your first attack fails, or if you do not elect to attack first, the war will occur. You had better hire some mercenaries since your enemy is doing the same. A mercenary is worth about 8 peasants in fighting power. Mercenaries cost 40 HL. each and there is a maximum of 75 mercenaries available to you. If your fighting power (mercenaries + peasants) exceeds your enemy's, you win; otherwise he wins. The winner acquires land from the loser in ratio to the size of the win. How much you fed the peasants last fall is now important and may occasionly make the difference between a win and a loss.

The winner also picks up some grain from the captured land and is able to harvest the captured land along with his own (at the same yield as his original land). The land acquired (or lost) will appear in next year's land quality table evenly distributed between the 100%, 80%, and 60% categories.

Since the mercenaries are horse mounted and the peasants are on foot, the mercenaries attack first. Thus, a large number of mercenaries will keep down your peasant losses whether you win or lose. The mercenaries must be paid after the battle. You can use granery reserves and the actual grain captured from acquired land (the one exception to the no-credit rule), but not the anticipated harvest (the mercenaries want their pay NOW).

If you can't pay all the mercenaries, they will attack your peasants, killing them and collecting grain from their huts until fully paid. Since the peasants don't have much grain left this late in the season, even a small default may cost you a lot of peasants. Incidentally, if the mercenaries do turn on the peasants, they also rape every female in the Duchy, making next year's birth rate very high. (We ignore the fact that the women deliver only a few months later -- these are no ordinary mercenaries.)

All peasant deaths from war cause resentment to build up against you. Attack by your own mercenaries is quite heavily resented.

Plagues and Poxes

The plague will kill off a third of the population, but in so doing it confers a 13 year immunity on the survivors. Therefore the plague cannot occur again for at least 13 years.

The pox is less deadly; it kills 10% or fewer peasants but confers no immunity. It can occur several years in a row.

Taxes and Expenses

The High King charges a tax of 1/2 HL. of grain for each HA. of land you possess (after war gains or losses). You had better be able to pay.

After the grain is harvested it must be milled. The castle granery can mill a maximum of 4000 HL. during the year. Additional harvest must be sent to the village miller at a charge of 10% of the amount milled. This amount is added to the castle overhead which is fixed at 120 HL. per year.

Births and Deaths

During the year, some natural deaths and numerous births have occured. Both are lumped together as if they occur just after the fall harvest.

The computer now prints out the results for the year and you start over again with the peasant's food decision.

Winning the Game

Through astute land management, profitable real-estate trading, winning a few wars and lots of luck, you may be able to build up your Duchy. If instead you let it decline, the High King may take it away from you and select a new manager. An unemployed Duke can find employment as a mercenary in somebody else's game.

Prosperity brings its risks. If you get too prosperous, the High King may become worried and begin to subsidize wars against you. These subsidies get larger as the game progresses.

If you should persevere, you may eventually beat some Duke so badly that you succeed in taking over his entire Duchy. In addition to the more than 400 HA. of land you will obtain, you get all of his surviving peasants (your war casualties will be positive) and the remaining contents of his granery. This poses a real threat to the crown and the High King will begin planning a direct attack against you. At the beginning of the following year the King will demand twice the usual tax. You may pay it and continue the game as usual, or you may refuse. You will never be rid of the double tax once it starts unless you refuse to pay it. This constitutes defiance of your Leige Lord and the King has his excuse for attacking you directly. The rest of the year will go as usual except that there will be no tax at all (no peasant levies either) and there will be no war threats (nobody dares).

The following year the King will attack just before planting time. You will have to hire as many foreign mercenaries as possible at 100 HL. each, grain in advance (the loser won't be in any position to pay). The program will automatically hire as many mercenaries as you can afford at the time. There is no limit to the number of foreign mercenaries you can hire except your current grain holdings. Each mercenary has as much fighting power as 8 peasants. If your total fighting strength is greater than the King's, you win. 250 to 300 mercenaries ought to be enough, depending on how many peasants you have.

Either way, the game is over. Good Luck!

Historical Waiver

No historical accuracy is implied in any way by this game. Except for the grain yields and planting requirements, the game is almost pure fiction. There were few mercenaries, Dukes did not often fight each other nor readily buy and sell land, the church was a power to be feared. The metric system had not yet been developed and the seven year locusts were not so reliable.

A Duke would have as his lord not a King but a Count or Earl and would have under him Barons or Marquises. There various nobles were the fighting force of the Kingdom (peasants did not fight). Taxes were paid not in grain but in periods of military service. (Yes, the National Guard was a medieval invention -- at the latest.)

```
        Credits

Original HAMMURABI game concept by Rick Merrill and David Ahl (1969)
KINGDOM program written by Lee Schneider and Todd Voros (1974) and
        distributed by Digital Equipment Corporation and, later,
        Data General Corporation
Original DUKEDOM program (loosely based on the
        KINGDOM program) written in D-level PL/I by
        Vince Talbot, Loyola Marymount University, Los Angeles
Rewritten (with extensive revision) in I.T.S. EXBASIC by
        Jamie E. Hanrahan
Adapted for /GAMES/ by David C. Barber
Re-written from I.T.S. EXBASIC to Hewlett-Packard level F BASIC
        then to DEC RSTS/E BASIC-PLUS then to Microsoft CP/M
        Disk BASIC by Richard A. Kaapke

(tm)    CP/M is a registered trademark of Digital Research, Inc.
        of Pacific Grove, Ca.
(tm)    EXBASIC is a trademark of International Timesharing Systems,
        Inc. (I.T.S.)
(tm)    RSTS/E and BASIC-PLUS are trademarks of Digital Equipment
        Corporation (DEC), of Maynard, Mass.
(C)     Microsoft BASIC is a proprietary product of Microsoft, Inc.
        of New Mexico.
(tm)    PL/I is a trademark of International Business Machines, Inc.
RUN

D U K E D O M
Microcomputer Version

Do you want instructions ? NO
Do you wish to skip the detailed reports at the end of each year ? NO


Year 0 Peasants 100 Land 600 Grain 4177

Peasants at start    96
Natural deaths       -4
Births                8
Peasants at end      100

Land at start        600
Land at end          600

100%  80%  60%  40%  20% Depl
 216  200  184    0    0    0

Grain at start       5193
Used for food       -1344
Seedings            -768
Crop yield           1516
Castle expense      -120
Royal tax           -300
Grain at end         4177

(Severe crop damage due to seven year locusts)


Grain for food = 15
Land to buy at 6 HL./HA. = 100
Land to plant = 400
Yield = 9.98 HL./HA.
Rats infest the grainery
The High King requires 3 peasants for his estates
and mines.  Will you supply them (Yes) or pay 300
HL. of grain instead (No) ? YES



Year 1 Peasants 101 Land 700 Grain 4629

Peasants at start    100
King's levy          -3
Natural deaths       -5
Births                9
Peasants at end      101

Land at start        600
Bought/sold          100
Land at end          700

100%  80%  60%  40%  20% Depl
 300  216  184    0    0    0

Grain at start       4177
Used for food       -1500
Land deals          -600
Seedings            -800
Rat losses          -169
Crop yield           3991
Castle expense      -120
Royal tax           -350
Grain at end         4629

Grain for food = 15
Land to buy at 17 HL./HA. = 0
Land to sell at 16 HL./HA. = 100
Land to plant = 400
Yield = 10.45 HL./HA.
Rats infest the grainery
The High King requires 2 peasants for his estates
and mines.  Will you supply them (Yes) or pay 200
HL. of grain instead (No) ? YES



Year 2 Peasants 103 Land 600 Grain 7185

Peasants at start    101
King's levy          -2
Natural deaths       -5
Births                9
Peasants at end      103

Land at start        700
Bought/sold          -100
Land at end          600

100%  80%  60%  40%  20% Depl
 200  300  100    0    0    0

Grain at start       4629
Used for food       -1515
Land deals           1600
Seedings            -800
Rat losses          -471
Crop yield           4180
Castle expense      -138
Royal tax           -300
Grain at end         7185


Grain for food = 15
Land to buy at 18 HL./HA. = 0
Land to sell at 17 HL./HA. = 100
Land to plant = 400
Yield = 8.1 HL./HA.
Rats infest the grainery
The High King requires 3 peasants for his estates
and mines.  Will you supply them (Yes) or pay 300
HL. of grain instead (No) ? Y


Year 3 Peasants 107 Land 500 Grain 8623

Peasants at start    103
King's levy          -3
Natural deaths       -5
Births                12
Peasants at end      107

Land at start        600
Bought/sold          -100
Land at end          500

100%  80%  60%  40%  20% Depl
 100  200  200    0    0    0

Grain at start       7185
Used for food       -1545
Land deals           1700
Seedings            -800
Rat losses          -787
Crop yield           3240
Castle expense      -120
Royal tax           -250
Grain at end         8623

Grain for food = 15
Land to buy at 15 HL./HA. = 0
Land to sell at 14 HL./HA. = 0
Land to plant = 400
Yield = 6.4 HL./HA.
Rats infest the grainery

Year 4 Peasants 112 Land 500 Grain 7659

Peasants at start    107
Natural deaths       -5
Births                10
Peasants at end      112

Land at start        500
Land at end          500
```

```
100%  80%  60%  40%  20% Depl
 100  100  200  100    0    0

Grain at start       8623
Used for food       -1605
Seedings            -800
Rat losses          -749
Crop yield           2560
Castle expense      -120
Royal tax           -250
Grain at end         7659

Grain for food = 154
The peasants demonstrate before the castle
with sharpened scythes
Grain for food = 15
Land to buy at 12 HL./HA. = 0
Land to sell at 11 HL./HA. = 0
Land to plant = 400
Yield = 6.75 HL./HA.
Rats infest the grainery

Year 5 Peasants 119 Land 500 Grain 6823

Peasants at start    112
Natural deaths      -5
Births               12
Peasants at end      119

Land at start        500
Land at end          500

100%  80%  60%  40%  20% Depl
   0  200  100  200    0    0

Grain at start       7659
Used for food       -1680
Seedings            -800
Rat losses          -686
Crop yield           2700
Castle expense      -120
Royal tax           -250
Grain at end         6823


Grain for food = 13
Land to buy at 13 HL./HA. = 0
Land to sell at 12 HL./HA. = 0
Land to plant = 470
Yield = 4.28 HL./HA.
Rats infest the grainery
A nearby Duke threatens war; Will you attack first ? NO
How many mercenaries will you hire at 40 HL. each = 40
You have won the war

Year 6 Peasants 121 Land 563 Grain 4059

Peasants at start    119
War casualties      -7
Natural deaths      -5
Births               14
Peasants at end      121

Land at start        500
Fruits of war        63
Land at end          563

100%  80%  60%  40%  20% Depl
  21   51  221  100  170    0

Grain at start       6823
Used for food       -1547
Seedings            -940
Rat losses          -574
Mercenary hire      -1600
Fruits of war        107
Crop yield           2191
Castle expense      -120
Royal tax           -281
Grain at end         4059


Grain for food = 14
Land to buy at 5 HL./HA. = 50
Land to plant = 450
Seven year locusts
Yield = 2.65 HL./HA.
A nearby Duke threatens war; Will you attack first ? YES
Peace negotiations were successful


Year 7 Peasants 122 Land 613 Grain 1981

Peasants at start    121
War casualties      -8
Natural deaths      -5
Births               14
Peasants at end      122

Land at start        563
Bought/sold          50
Land at end          613

100%  80%  60%  40%  20% Depl
   0   21  214  271  100    7

Grain at start       4059
Used for food       -1694
Land deals          -250
Seedings            -900
Crop yield           1192
Castle expense      -120
Royal tax           -306
Grain at end         1981


Grain for food = 10
The peasants demonstrate before the castle
with sharpened scythes
Grain for food = 1981
Land to buy at 4 HL./HA. = 0
Land to sell at 3 HL./HA. = 100
The High King appropiates half your earnings
as punishment for selling at such a low price
Land to plant = 400
But you don't have enough grain
You have 150 HL. of grain left,
Enough to plant 75 HA. of land
Land to plant = 75
Yield = 7.21 HL./HA.


Year 8 Peasants 136 Land 513 Grain 164

Peasants at start    122
Natural deaths      -6
Births               20
Peasants at end      136

Land at start        613
Bought/sold         -100
Land at end          513

100%  80%  60%  40%  20% Depl
  60  271  121   61    0    0

Grain at start       1981
Used for food       -1981
Land deals           150
Seedings            -150
Crop yield           540
Castle expense      -120
Royal tax           -256
Grain at end         164


The peasants tire of war and starvation
You are deposed
LIST
1000    CLEAR 400:
        '
        '               D U K E D O M  --  MICROCOMPUTER VERSION
        '
1002    DEF FNRX(Q1%, Q2%) = RND(1) * (1 +Q2% -Q1%) +Q1%
1003    DEF FNXX(Q1%) = FNRX(-F3%, F3%) + R(Q1%)
1010    GOTO 1620       ' SKIP SUBROUTINE DEFINITIONS
        '
        '       DECLARATIONS AND SUBPROGRAM DEFINITIONS
        '
1030    DIM P(8), L(3), G(10), S(6), U(6), R(8), P$(8), L$(3), G$(10)
1040    '       ---PARTIALLY GAUSSIAN RANDOM NUMBERS
1050    Q3% = FNRX(Q1%, Q2%):
        IF FNRX(Q1%, Q2%) > 5 THEN GO% = (Q3% +FNRX(Q1%, Q2%))/2
           ELSE GO% = Q3%
1060    RETURN
        '       ---READ A YES-OR-NO RESPONSE
1330    LINE INPUT V$ : V$ = LEFT$(V$, 1):
        IF LEN(V$)=0 THEN V$="N" ELSE
        IF ASC(V$) > 95 THEN V$ = CHR$(ASC(V$)-32)
1332    IF V$="N" OR V$="Y" THEN RETURN ELSE
        PRINT "Please answer Yes or No: ";:  GOTO 1330
        '       ---READ A NUMERIC RESPONSE
1430    LINE INPUT V$ : V = INT(VAL(V$)):
        IF V >=0 THEN RETURN ELSE
```

```
      PRINT "Please enter a non-negative number: ";:  GOTO 1430
      '       ---COMMON ERROR MESSAGES
1530  PRINT "But you don't have enough grain":
      PRINT "You have" G "HL. of grain left,":
      IF X1>=4 THEN PRINT "Enough to buy" INT(G/X1) "HA. of land"
1531  PRINT "Enough to plant" INT(G/2) "HA. of land":
      RETURN
1550  PRINT "But you don't have enough land":
      PRINT "You only have" L "HA. of land left":
      RETURN
1570  PRINT "But you don't have enough peasants":
      PRINT "Your peasants can only plant" 4*P "HA. of land":
      RETURN
      '
      '       INTRODUCTION TO PROGRAM
      '
1620  PRINT: PRINT "D U K E D O M": PRINT "Microcomputer Version":
      PRINT: PRINT "Do you want instructions ? ";:  GOSUB 1330:
      IF V$="N" THEN 2050
1690  OPEN "I", 2, "DUKEDOM.INS"
1700  LINE INPUT #2, V$
1701  IF LEFT$(V$,5)<>"*EOF*" THEN PRINT V$:  GOTO 1700
1710  PRINT "Do you require a more detailed description
      (it's LONG) ? ";:  GOSUB 1330:
      IF V$="N" THEN 2050
1720  IF EOF(2) THEN 2050 ELSE
      LINE INPUT #2, V$:  PRINT V$:  GOTO 1720
      '       ---ONCE ONLY INTIIALIZATION
2050  CLOSE 2:
      F3% = 2:  M = 1.95
      '
      '       START NEW GAME
      '
2100  PRINT "Do you wish to skip the detailed reports at the end
      of each year ? ";: GOSUB 1330:  R$ = V$
2180  READ Y%, C1, U1, U2, K%, D, P, L, G
2190  FOR I% = 1 TO 8:  READ P(I%):  NEXT:
      FOR I% = 1 TO 3:  READ L(I%):  NEXT:
      FOR I% = 1 TO 10: READ G(I%):  NEXT:
      FOR I% = 1 TO 6:  READ S(I%):  NEXT:
      FOR I% = 1 TO 8:  READ P$(I%):  NEXT:
      FOR I% = 1 TO 3:  READ L$(I%):  NEXT
2195  FOR I% = 1 TO 10:  READ G$(I%):  NEXT
      ' AND CURSE THIS BASIC FOR NOT HAVING MAT FUNCTIONS '
2200  RESTORE
2210  DATA    0, 3.95, 0, 0, 0, 0, 100, 600, 4177,
              96, 0, 0, 0, 0, 0, -4, 8,
              600, 0, 0,
              5193, -1344, 0, -768, 0, 0, 0, 1516, -120, -300,
              216, 200, 184, 0, 0, 0
2260  '
      '       INITIALIZE RANDOM BASE TABLE
      '
2270  Q1% = 4:  Q2% = 7:  GOSUB 1050:  R(1) = G0%:
      Q2% = 8:  GOSUB 1050:  R(2) = G0%:
      Q2% = 6:  GOSUB 1050:  R(3) = G0%:
      Q1% = 3:  Q2% = 8:  GOSUB 1050:  R(4) = G0%:
      Q1% = 5:  GOSUB 1050:  R(5) = G0%:
      Q1% = 3:  Q2% = 6:  GOSUB 1050:  R(6) = G0%
2271  Q2% = 8:  GOSUB 1050:  R(7) = G0%:
      Q1% = 4:  GOSUB 1050:  R(8) = G0%
      '
      '       DISPLAY LAST YEAR'S RESULTS
      '
2460  PRINT: PRINT: PRINT "Year" Y% "Peasants" P "Land" L "Grain"
      G: PRINT
2500  IF R$="Y" THEN 2810      ' SKIP LONG REPORT IF NOT WANTED

2510  FOR J1% = 1 TO 8:
         IF P(J1%)<>0 OR J1%=1 THEN PRINT P$(J1%); TAB(20); P(J1%)
2520  NEXT J1%:
      PRINT "Peasants at end"; TAB(20); P: PRINT
2600  FOR J1% = 1 TO 3:
         IF L(J1%)<>0 OR J1%=1 THEN PRINT L$(J1%); TAB(20); L(J1%)
2610  NEXT J1%:
      PRINT "Land at end"; TAB(20); L: PRINT
2690  PRINT "100%  80%  60%  40%  20% Depl":
      PRINT USING "####"; S(1);:
      FOR J1% = 2 TO 6:
         PRINT USING "#####"; S(J1%);:
      NEXT J1%:  PRINT:  PRINT
2720  FOR J1% = 1 TO 10:
         IF G(J1%)<>0 OR J1%=1 THEN PRINT G$(J1%); TAB(20); G(J1%)
2730  NEXT J1%:
      PRINT "Grain at end"; TAB(20); G: PRINT
2800  IF Y%<=0 THEN PRINT "(Severe crop damage due to seven year locusts)
2810  PRINT: PRINT:  Y% = Y%+1:
      FOR I% = 1 TO 8:  P(I%) = 0:  NEXT:
      FOR I% = 1 TO 3:  L(I%) = 0:  NEXT:
      FOR I% = 1 TO 10: G(I%) = 0:  NEXT
2815  P(1) = P:  L(1) = L:  G(1) = G
      '
      '       TEST FOR END-OF-GAME CONDITIONS
      '
2980  IF P >= 33 THEN 3040 ELSE
      PRINT "You have so few peasants left that"
3000  PRINT "the High King has abolished your Ducal right":
      PRINT:  GOTO 30000
3040  IF L >= 199 THEN 3070 ELSE
      PRINT "You have so little land left that":  GOTO 3100
3070  IF U1 > 88 OR U2 > 99 THEN 3100 ELSE
      IF G >= 429 THEN 3130
3100  PRINT "The peasants tire of war and starvation":
      PRINT "You are deposed":  PRINT:  GOTO 30000
3130  IF Y% > 45 AND K% = 0 THEN
      PRINT "You have reached the age of mandatory retirement":
      GOTO 30000
3170  U1 = 0:
      IF K% > 0 THEN PRINT "The King demands twice the royal tax
         in the":
         PRINT "hope of provoking war.  Will you pay ? ";:
         GOSUB 1330:  K% = 2:
         IF V$="N" THEN K% = -1
      '
      '       FOOD FOR THE PEASANTS
      '
3270  PRINT "Grain for food = ";:
      GOSUB 1430 : V = -V*(V>=100) -V*P*(V<100):
      IF V > G THEN GOSUB 1530:  GOTO 3270
3350  IF V/P < 11 AND V<>G THEN
         PRINT "The peasants demonstrate before the castle":
         PRINT "with sharpened scythes":  U1 = U1+3:  GOTO 3270
3410  G(2) = -V:  G = G+G(2)
      '       ---STARVATION AND UNREST CALCULATIONS
3440  X1 = V/P:
      IF X1 < 13 THEN PRINT "Some peasants have starved during the
      winter"
      P(2) = -INT(P-V/13):  P = P+P(2)
3490  X1 = X1-14:  X1 = -X1*(X1<=4) -4*(X1>4):
      U1 = U1 -3*P(2) -2*X1:
      IF U1 > 88 THEN 3100 ELSE
      IF P<33 THEN 2980
      '
      '       PURCHASE OR SALE OF LAND
      '
3580  C = C1:  X1 = INT(2*C+FNX%(1)-5):
      X1 = -X1*(X1>=4) -4*(X1<4)
3640  PRINT "Land to buy at" X1 "HL./HA. = ";:
      GOSUB 1430:  G(3) = -V*X1:
      IF -G(3) > G THEN GOSUB 1530:  GOTO 3640
3700  L(2) = V:  S(3) = S(3) + V:
      IF V>0 THEN 4020 ELSE X2 = S(1)+S(2)+S(3)
3740  FOR J1% = 1 TO 3:
         X1 = X1-1:
         PRINT "Land to sell at" X1 "HL./HA. = ";:
         GOSUB 1430:
         IF V>X2 THEN PRINT "But you only have" X2 "HA. of good land":
            GOTO 3840
3810  G(3) = V*X1:
      IF G(3) <= 4000 THEN 3870 ELSE
      PRINT "No buyers have that much grain - sell less"
3840  NEXT J1%:  PRINT "Buyers have lost interest":
      V = 0:  G(3) = 0
3870  L(2) = -V
3880  FOR J1% = 3 TO 1 STEP -1:
         IF V <= S(J1%) THEN 4010 ELSE
         V = V-S(J1%):  S(J1%) = 0:
      NEXT J1%
      ' LOOP SHOULD NEVER EXIT HERE '
3960  PRINT "LAND SELLING LOOP ERROR - CONTACT PROGRAM AUTHOR IF":
      PRINT "ERROR IS NOT YOURS IN ENTERING PROGRAM,":
      PRINT "AND SEEMS TO BE FAULT OF PROGRAM'S LOGIC.": STOP
4010  S(J1%) = S(J1%) -V

4020  L = L+L(2):
      IF L<10 THEN 3040 ELSE
      IF L(2)<0 AND X1<4 THEN G(3) = INT(G(3)/2):
         PRINT "The High King appropiates half your earnings":
         PRINT "as punishment for selling at such a low price"
4090  G = G+G(3)
      '
      '       WAR WITH THE KING
      '
4130  IF K%<> -2 THEN 4350 ELSE
      PRINT "The King's army is about to attack your duchy":
      X1 = INT(G/100):
      PRINT "at 100 HL. each (payment in advance)"
4170  PRINT "you have hired" X1 "foreign mercenaries":
      IF 8*X1+P>2399 THEN 4220
4190  PRINT "The placement of your head atop the castle gate":
      PRINT "signifies that ";:  GOTO 3000
4220  PRINT "Wipe the blood from the crown - you are now High King!":
      PRINT: PRINT "A nearby monarchy threatens war; ";:
      PRINT "how many .....":PRINT:PRINT:PRINT:GOTO 32767
      '
      '       GRAIN PRODUCTION
      '
```

```
4350    PRINT "Land to plant = ";:  GOSUB 1430:
        IF V>L THEN GOSUB 1550:  GOTO 4350
4360    IF V>4*P THEN GOSUB 1570:  GOTO 4350
4430    G(4) = -2*V:
        IF -G(4) > G THEN GOSUB 1530:  GOTO 4350
4470    G(8) = V:  G = G+G(4)
        '       ---UPDATE LAND TABLES
4500    FOR I% = 1 TO 6:  U(I%) = 0:  NEXT
4510    FOR J1% = 1 TO 6:
           IF V<=S(J1%) THEN 4620 ELSE
           V = V-S(J1%):  U(J1%) = S(J1%):
           S(J1%) = 0:
        NEXT J1%
        '       LOOP SHOULD NEVER REACH THIS FAR
4600    PRINT "LAND TABLE UPDATING ERROR - PLEASE CONTACT PROGRAM AUTHOR":
        PRINT "IF ERROR IS NOT A FAULT OF ENTERING THE PROGRAM, BUT RATHER":
        PRINT "FAULT OF THE PROGRAM LOGIC.": STOP
4620    U(J1%) = V:  S(J1%) = S(J1%) - V:
        S(1) = S(1)+S(2):  S(2) = 0
4660    FOR J1% = 3 TO 6:
           S(J1% -2) = S(J1%-2)+S(J1%):
           S(J1%) = 0:
        NEXT J1%
4700    FOR J1% = 1 TO 5:  S(J1%+1) = S(J1%+1)+U(J1%):  NEXT:
        S(6) = S(6)+U(6)
        '
        '       CROP YIELD AND LOSSES
        '
4770    C = FNX%(2)+3:
        IF INT(Y%/7) * 7 = Y%  THEN PRINT "Seven year locusts":
        C = C/2
4830    X1 = 0:
        FOR J1% = 1 TO 5:
           X1 = X1 + U(J1%) * (1.2 - .2*J1%):
        NEXT J1%:
        IF G(8)=0 THEN C1 = 0:  C = 0:
        ELSE C1 = INT((C *(X1/G(8)))*100)/100:  C = C1
4920    PRINT "Yield =" C "HL./HA.":
        X1 = FNX%(3)+3:
        IF X1 < 9 THEN 5190 ELSE G(5) = -INT((X1*G)/83):
        G = G+G(5)
4990    PRINT "Rats infest the grainery":
        IF P<67 OR K%=-1 THEN 5190 ELSE
        X1 = FNX%(4):
        IF X1> P/30 THEN 5190
5060    PRINT "The High King requires" X1 "peasants for his estates":
        PRINT "and mines.  Will you supply them (Yes) or pay" X1*100:
        PRINT "HL. of grain instead (No) ? ";:  GOSUB 1330:
        IF V$="N" THEN G(10) = -100*X1:  G = G+G(10):  GOTO 5190
5140    P(3) = -X1:  P = P+P(3)
        '
        '       W A R
        '
5190    IF K%=-1 THEN PRINT "The High King calls for peasant levies":
           PRINT "and hires many foreign mercenaries":  K% = -2:  GOTO 6340
5240    X1 = INT(11 - 1.5*C):
        X1 = -X1*(X1>=2) -2*(X1<2):
        IF K%<>0 OR P<=109 OR 17*(L-400)+G <= 10600 THEN 5350
        ELSE PRINT "The High King grows uneasy and may"
5310    PRINT "be subsidizing wars against you":  X1 = X1 +2:
        X2 = Y%+5:  GOTO 5380
5350    X2 = 0
5380    X3 = FNX%(5):
        IF X3>X1 THEN 6340 ELSE
        PRINT "A nearby Duke threatens war; ";:
        X2 = INT(X2+85+18*FNX%(6))
5440    X4 = 1.2-U1/16:  X5 = INT(P*X4)+13:
        PRINT "Will you attack first ? ";:  GOSUB 1330:
        IF V$="N" THEN 5590 ELSE
        IF X2>=X5 THEN 5540 ELSE
        PRINT "Peace negotiations were successful"
5510    P(4) = -X1-1:  X2 = 0:  GOTO 5570
5540    PRINT "First strike failed - you need professionals":
        P(4) = -X3-X1-2:  X2 = X2+3*P(4)
5570    P = P+P(4):
        IF X2<1 THEN U1 = U1-2*P(4)-3*P(5):  GOTO 6340
5590    PRINT "How many mercenaries will you hire at 40 HL. each = ";:
        GOSUB 1430:
        IF V>75 THEN
           PRINT "There are only 75 mercenaries available for hire":
           GOTO 5590
5640    X2 = INT(X2*M):  X5 = INT(P*X4)+7*U+13:
        X6 = X2-4*U-INT(.25*X5):  X2 = X5-X2:
        L(3) = INT(.8*X2):
        IF -L(3)>.67*L THEN
           PRINT "You have been overrun and have lost the entire Dukedom":
           GOTO 4190
5720    X1 = L(3)
5730    FOR J1% = 1 TO 3:
           X3 = INT(X1/(4-J1%)):
           IF -X3 <= S(J1%) THEN X5 = X3 ELSE X5 = -S(J1%)
5790       S(J1%) = S(J1%) + X5:
           X1 = X1-X5:
        NEXT J1%
5820    FOR J1% = 4 TO 6:
           IF -X1 <= S(J1%) THEN X5 = X1 ELSE X5 = -S(J1%)
           S(J1%) = S(J1%) + X5:
           X1 = X1-X5:
        NEXT J1%
5900    IF L(3)<399 THEN 6010 ELSE
        PRINT "You have overrun the enemy and annexed his entire Dukedom":
        G(7) = 3513:  G = G+G(7):  X6 = -47:  X4 = .55:
        IF K%>0 THEN 6130
5970    K% = 1:
        PRINT "The King fears for his throne and":
        PRINT "may be planning direct action":  GOTO 6130
6010    IF X2>=0 THEN
           PRINT "You have won the war":  X4 = .67:
           G(7) = INT(1.7*L(3)) : G = G+G(7):  GOTO 6090
6070    PRINT "You have lost the war":  X4 = G(8)/L
6090    IF X6<=9 THEN X6 = 0 ELSE X6 = INT(X6/10)
6130    X6 = -X6*(X6<=P) -P*(X6>P):  P(4) = P(4)-X6:
        P = P-X6:  G(8) = G(8)+INT(X4*L(3)):
        X6 = 40*V:
        IF X6 <= G THEN G(6) = -X6 ELSE G(6) = -G:
           P(5) = -INT((X6-G)/7)-1: PRINT "There isn't enough grain to
           pay the mercenaries"
6250    G = G+G(6):
        P(5) = -P(5)*(-P(5)<=P) +P*(-P(5)>P):
        P = P+P(5):
        L = L+L(3):  U1 = U1-2*P(4)-3*P(5)
        '
        '       PLAGUES, BIRTHS, NATURAL DEATHS
        '
6340    X1 = FNX%(7):
        IF X1>3 THEN 6500 ELSE
        IF X1<>1 THEN 6440 ELSE
        IF D>0 THEN 6500 ELSE
        PRINT "The BLACK PLAGUE has struck the area":
        D = 13:  X2 = 3
6430    P(6) = -INT(P/X2):  P = P+P(6):  GOTO 6500
6440    PRINT "A POX EPIDEMIC has broken out":  X2 = X1*5:
        P(6) = -INT(P/X2):  P = P+P(6)
6500    X1 = FNX%(8)+4:
        X1 = -X1*(P(5)=0) -4.5*(P(5)<>0):  P(8) = INT(P/X1):
        P(7) = INT(.3-P/22):  P = P+P(7)+P(8):  D = D-1
        '
        '       GRAIN HARVEST, CASTLE EXPENSES, KING'S TAX
        '
6600    G(8) = INT(C*G(8)):  G = G+G(8):  X1 = G(8)-4000:
        G(9) = -G(9)*(X1<=0) +INT(.1*X1)*(X1>0):  G(9) = G(9)-120:
        G = G+G(9)
6670    IF K%<0 THEN 6790 ELSE X1 = -INT(L/2)
6710    X1 = -X1*(K%<2) -2*X1*(K%>=2):
        IF -X1>G THEN
           PRINT "You have insufficient grain to pay the royal tax":
           GOTO 3000
6740    G(10) = G(10)+X1:  G = G+X1
        '
        '       UPDATE UNREST COUNTER AND CONTINUE
        '
6790    U2 = INT(U2*.85)+U1:  GOTO 2460
7000    DATA    "Peasants at start", "Starvations", "King's levy",
                "War casualties", "Looting victims", "Disease victims",
                "Natural deaths", "Births"
7100    DATA    "Land at start", "Bought/sold", "Fruits of war"
7200    DATA    "Grain at start", "Used for food", "Land deals",
                "Seedings", "Rat losses", "Mercenary hire",
                "Fruits of war", "Crop yield", "Castle expense",
                "Royal tax"
30000   PRINT:PRINT "Do you wish to try another game ? ";:  GOSUB 1330:
        IF V$="Y" THEN 2180
32767   END '
Ok
```

Computer Simulations in the Classroom

Larry Noonan

A computer simulation is a system designed to reproduce essential aspects of reality for the purpose of finding ways to manage, control, solve, and/or ultimately agree upon the best solution for a problem.

In the classroom a simulation allows students to make judgments and, ultimately, decisions based on logic, observation and their knowledge of the real world. If the simulation is also a game, as is the computer simulation that I will discuss here, then it becomes an educational game. In the classroom, an educational game can change passive learning into an active experience that helps students understand more easily the concepts being taught.

Some Questions

Why use a computer simulation? This is a very important question. Computer simulations are not meant to replace hands-on activities and experiments, but to supplement them. Before using a computer simulation in the classroom you should ask these questions.

1. Can the simulation that you are considering be done better in a different way? If the answer to this question is "yes" then you should use that alternative and forget about computer simulation.

2. Does the computer simulation make good use of the capabilities of the computer? Does it use graphics, simplify record keeping, eliminate unnecessary calculations, etc.? Graphics help the student to visualize the experiment and simulate work with real equipment and materials. If the student or teacher must do all the calculations and keep all the results and records for the experiment or demonstration, then it might be better to do the whole simulation using paper and pencil only. Some record keeping may, however, be desirable for comparison purposes as we shall see later.

If, after you have answered these two questions, you feel that a computer simulation is what you want, consider the following four reasons that I often use to justify my use of this particular teaching tool.

Larry Noonan, 787 Strouds Lane, Pickering, Ontario, Canada.

1. The students are immediately turned on to this machine which has a certain amount of control over their future in the game.

2. The feedback or effects of their decisions are seen very quickly.

3. The simulation can be more complicated in its calculations (e.g. sales tax influences jobs which influence income tax, etc.) while keeping the outward appearance as simple as possible.

4. The random portion of the game or simulation is controlled by an unbiased machine.

Other reasons that computer simulations are enjoying increasing popularity are related to the fact that some things cannot be done in any other way. For example, some experiments and demonstrations happen too fast to observe. Others, like heredity experiments, may take too long. A computer simulation can slow down or speed up these experiments so that the results and the processes are easily observed. Some experiments such as those involving high voltage or radioactive materials are too dangerous to do in the classroom but can be safely simulated. Experiments involving expensive or difficult-to-obtain materials and apparatus can usually be simulated on a computer for a fraction of the cost of the actual experiment. Some experiments are too complicated. A computer can simplify these experiments so that only the important concepts and their interactions are seen.

Sources of Computer Simulations

Most software distributors advertise computer simulations. Some are educational but most are classified as entertainment. There are two ways in which many of these can be adapted for the classroom.

1. Make a commercial program fit into the classroom situation and develop activities and objectives that happen to fit that program.

2. Try to find a simulation that accomplishes the educational objectives which you wish to teach. This method is often more difficult but it is, in my opinion, better to fit the simulation to the needs of the class than vice versa. Therefore, I have chosen to use this method in my classroom.

Educational Objectives

There are two kinds of educational objectives. The first, referring to the knowledge that should be attained by the end of the simulation, is called a "target objective." The second refers to those skills which are acquired during the course of simulation because they are necessary to meet the target objectives. These are called the "enabling objectives."

My target objective for the computer simulation was: to have the student, given a series of simulated real life situations, recognize and identify the cause and effect relationships involved by trying different solutions to the problems presented by the computer.

The program that I chose to achieve this objective is *Santa Paravia and Fiumaccio* for the TRS-80 Level II from Instant Software. This program has very little documentation, none of which refers to educational uses. It allows teams or groups of students to participate simultaneously and keeps records for up to six groups.

The game crowns a ruler whose objective is to accumulate land, grain and tax money and to construct buildings in his city while keeping his subjects alive and, presumably, happy.

The objectives in the program correspond with my target objective because the game requires the problem solving and intellectual skills of identifying cause and effect relationships. It also involves the cognitive strategy of originating solutions to such problems as what to do when bad weather strikes and how to balance revenues with grain production to keep the people healthy and happy. The students must try different methods to solve these and other problems, study the results and then apply what they have learned to create new and better solutions to the problems.

My enabling objective for this simulation was: to have the student, given a group situation simulating society, demonstrate a co-operative attitude within a group by helping to make the decisions that would lead to successful completion of the game.

This objective requires students to work in a group for the common good of the group. Without co-operation within the group, failure to complete the game successfully is all but certain. It was hoped that each student in the group would see co-operation as an essential means to the desired end.

Intellectual skills required to calculate the best way to proceed and practice in decision making are also among the enabling objectives.

To use this simulation the students must have certain skills. They are:

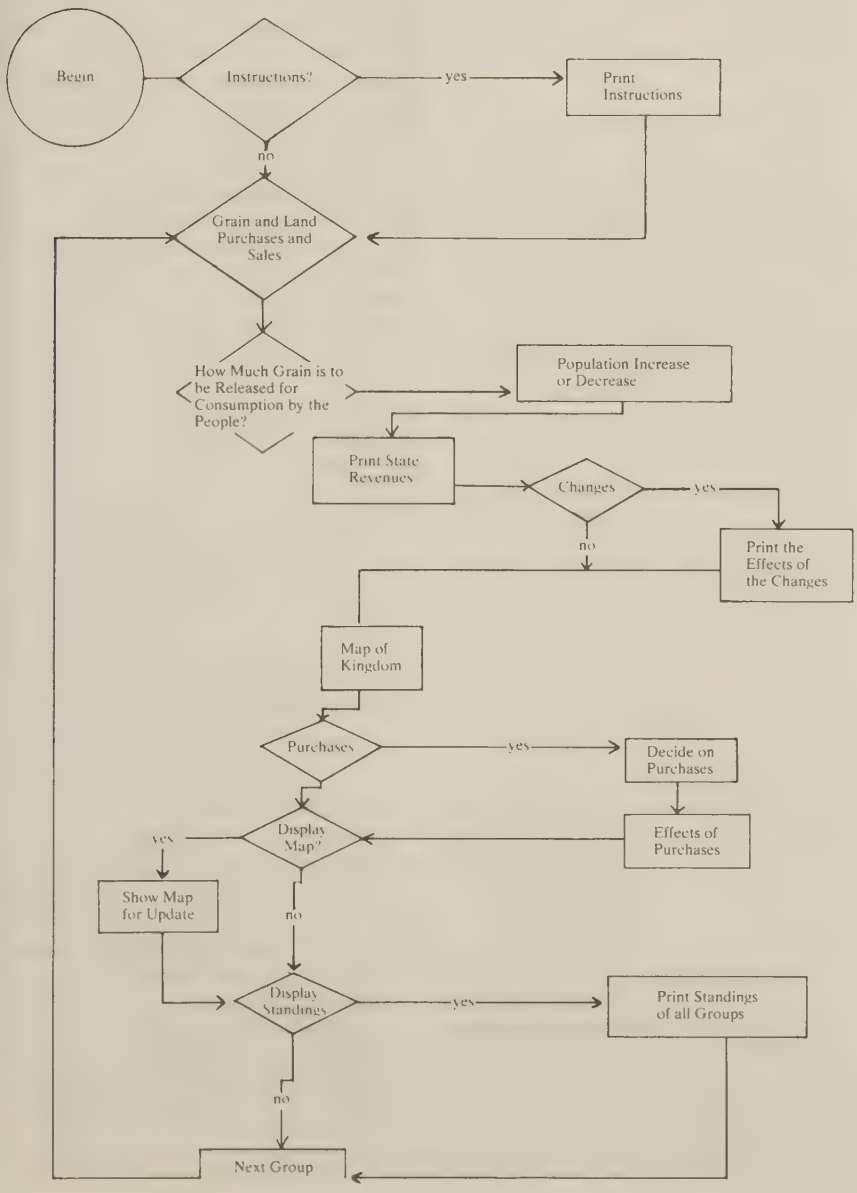
- •Computational skills of addition, subtraction, multiplication and division
- •The ability to read with understanding
- •The motor skills of drawing straight and curved lines
- •The ability to write
- •Mapping skills

Figure 1 shows a flow chart of the simulation.

Figure 1. Flow Chart.

Diamonds indicate areas where the students must make decisions.

Rectangles indicate what the computer displays on the video screen.



Using a flow chart allows the teacher to examine and understand the program and devise worksheets for the students.

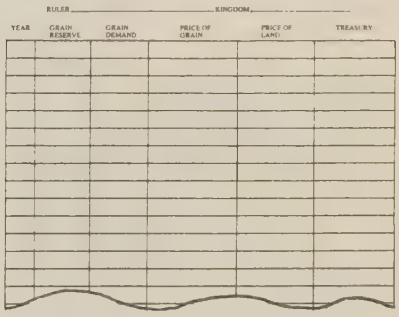
Worksheets

As a permanent record for the students and the teacher it is a wise idea to develop worksheets that will be filled in during the running of the program. As already mentioned, these worksheets should not require a large number of calculations. To draw up the worksheets, examine the simulation and find all the areas where the computer reports results or observations.

I developed worksheets in the following areas.

1. *Grain and Land* (See Figure 2.) The groups must decide to sell or buy grain on the basis of crop yield for that year, the price of grain and the amount needed to feed the people. Selling grain helps the treasury to buy markets and factories to make more money and create jobs. If too much is sold, however, there may not be enough to feed the people, and starvation will result. Fewer people means fewer tax payers which means less money for the treasury. Land can be purchased on which to grow more grain or it can be sold for cash.

Figure 2. Grain and Land.



2. *State Revenues* (See Figure 3.) The taxes and duties can be increased or lowered affecting the economy of the nation. For example, if the sales tax is raised, fewer products are purchased and fewer people are required to produce the products. As a result, the total income tax is decreased. (All of the complicated calculations are done by the computer.)

3. *Standings* (See Figure 4.) This sheet allows the groups to study the progress of other groups, compare their own performance and decide where they need to improve. It helps them to decide on strategy.

4. *Map* (See Figure 5.) This map is updated with every turn (every year in the game) and helps the groups to visualize their kingdoms.

Figure 3. State Revenue.

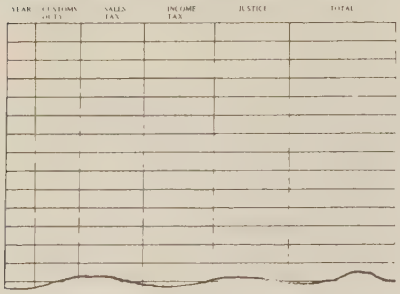


Figure 4. Standings.

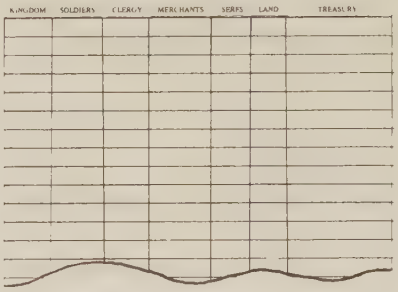


Figure 5. Map.

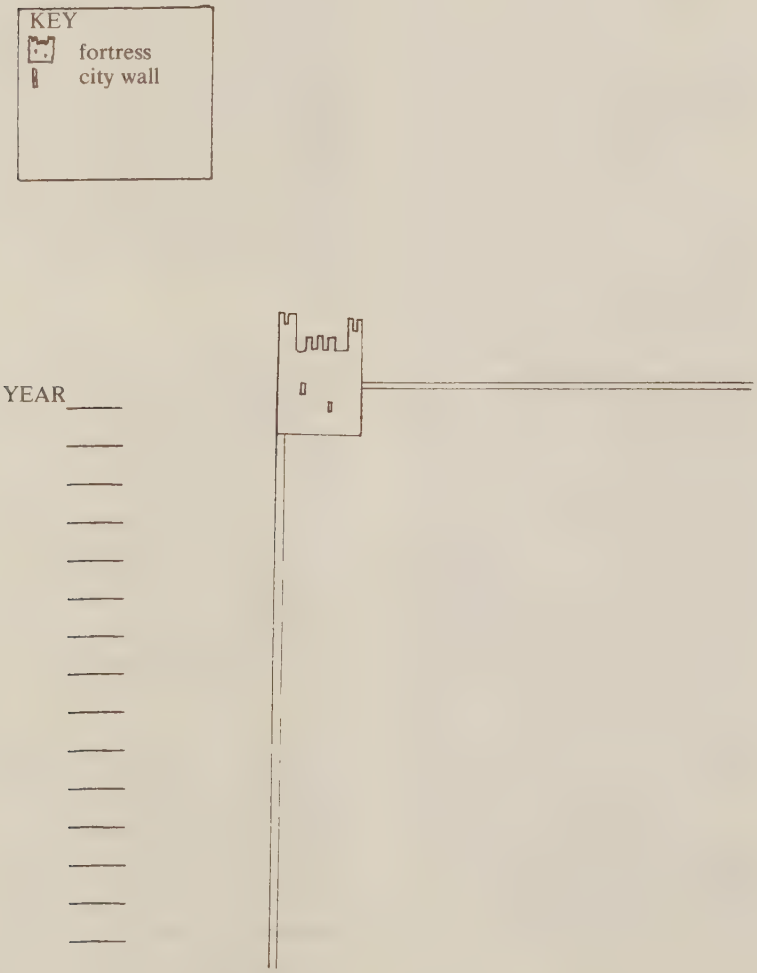


Figure 6. Learner Flow Chart.

1. DECISION - buy or sell land
2. DECISION - buy or sell grain
3. DECISION - amount of grain to give to the people
4. DECISION - raise or lower or keep taxes the same
5. DECISION - build markets, etc. or not build them

Learner Flow Chart

In order to analyze the possible results of a decision a learner flow chart should be drawn and examined. These flow charts can be as simple as the one in Figure 6.

The five decisions that the students make during each turn, all have a cause and effect relationship that the student hopefully will soon recognize and then use in further decisions. Some of the effects are immediate (e.g., if the people receive too little food many of them starve. This, of course is to be avoided.) Other effects take longer to see (e.g., buying land results in the production of more grain during the next year).

Analysis of Learner Flow Chart

1. Decisions are made concerning purchases and sale of land. The students that used this simulation quickly realized that more land meant more crops while less land meant more immediate money. These facts are weighed one against the other and a decision is made. The effect of the decision is apparent during the next turn.

2. A decision must be made concerning the buying or selling of grain. If there is too much grain because of a good crop year then the students usually sell some of it for money that can be used in other areas. If there has been a bad year, I found that the students had more difficulty making a choice to buy or to let some of their people starve. Since the object of the game is to make as many gains as possible, including population, the usual decision was to buy grain. I observed during this process that the majority of students were concerned about keeping as much money as possible in order to buy markets and make more money. There was however, in each group, at least one person who saw the results of giving less food to the people (i.e., starvation). It was this one person or small group within the group that always prevailed. These people, although usually not the dominant players on their teams, used good arguments to convince the rest of the group. I found that good judgment consistently won out over apparent greed. I was surprised to find this.

3. Calculating skills were required to make the decision of how much grain to give to the people. Some grain was needed for the next year. Too little grain given to the people meant immediate death for many, while too much meant that babies would be born and immigration would increase. Too much grain was the usual choice at the beginning of the game until a year of bad weather occurred, the supply of grain was exhausted, and many serfs died. From then on the students took considerably more time in calculating their grain reserves.

Cause and effect relationships were seen

most easily here as people starved for lack of food or births and immigration increased because of an abundance of food.

Here as in the other decisions, the students evaluated previous decisions and their effects and tried to improve upon them during subsequent turns.

4. Decisions must be made about raising or lowering taxes. Results are, again, immediate, however there are long term effects which are less evident. An example would be that a decrease in sales tax produces immediate loss of revenue but stimulates production and increases jobs so that income tax revenues increase. This second result is not evident until subsequent turns. I found that most students had difficulty recognizing this point.

5. The fifth decision involves the building of markets and mills. To build costs money but creates more jobs and taxes. If the following year has good weather then everything will be fine for those groups that spend all their money on building projects. If bad weather and poor crops come, then money is needed to buy grain or many will starve. It is here where the element of chance comes in.

The students usually took one of the following routes:

1. Gamble on good weather and deplete the treasury to build markets. These groups often made quick gains but rarely ended up in the lead when the game was over.

2. Build fewer markets and mills and keep some money in reserve. The groups that chose this course made slower progress but usually ended up higher than the gambling groups at the conclusion of the game.

Conflicts often arose within the groups as to which way to go. The attitude toward working in a group for the common good of the group is developed during situations like this. If no decision could be agreed on then the leader of the group had the final say. The most important thing for me, as the teacher, to ensure was that the decision (cause) was made and its results (effects) were seen and studied.

Testing Results

When using a computer simulation in the classroom, it is difficult if not impossible to test formally the results of the learning. Instead, the final outcome and observations of the groups form the basis of the testing that must be done. By observing the groups and how they work it becomes clear which groups and which individuals are developing the necessary co-operative attitude which is so essential in the game. Periodic examination of worksheets will also help to determine if the students are learning from their experiences with cause and effect relationships.

If, when the game is over, the group has more money, population, markets, etc. than when it started, it is deemed successful. The degree of success is measured in the final difference between that with which it started and that with which it ended. If, for instance, two groups start out with 1000 people and one ends up with 10,000 while the other ends up with 2000 then the first group understands the cause and effect relationships in the game better than does the second group which could have increased its population by pure luck.

By examining all the areas where gains can be made and realizing that the greater the gains the more likely the group understood the purpose of the game and the more likely that they could recognize and work with cause and effect relationships, you can evaluate each group.

If a group ends up with less than they started with or with only a small gain, then it is likely that cause and effect relationships have not been fully understood.

Introducing the Computer Simulation to the Class

Now that you have chosen the simulation, thought about your educational objectives, developed worksheets and become thoroughly familiar with the computer program itself, how do you introduce it to your students?

The term simulation is heard on TV and in the movies enough so that it can be briefly explained to the students as a game that allows a person to make judgments and decisions based on past experience and their knowledge of the real world. A discussion should follow this definition so that the students really understand what they will be doing.

The students should be divided into groups of up to six and the class arranged so that the groups can sit together from the start. This will hopefully promote the co-operative attitude that is so important to the game.

Each of the worksheets that the groups will be using can be made into overhead projections and these can be used to explain the basic rules of the game.

Using each of the worksheets in turn the teacher can work through the techniques of entering data and making decisions. The possible effects of certain actions could be discussed. Examples:

1. What might happen if you sold almost all of your grain?

2. What might happen if you raised the sales tax to 100% to make an object cost twice its worth?

When the basic rules are understood, each group should select a leader. The leader is then asked to select a person or persons for each of the following positions.

•Department of Agriculture: This person uses the Grain and Land Worksheet.

•Treasurer: This person uses the State Revenue Worksheet.

•Department of Commerce: This person uses the State Revenue Worksheet.

•Department of Lands: This person uses the Grain and Land Worksheet.

Results

When I used this computer simulation in my classes (grades 3 to 7) I observed several things. Here are just a few.

Around the fifth turn I noticed that some groups came to the computer prepared to enter values they had already calculated at their desks. This was a good indication that they were looking at the results of their past decisions and making new choices based on the observed effects.

Each time I ran the simulation, some groups hastily made decisions for their first four or five turns, a practice which caused them to fall behind the other groups. As the game went on I noticed that all of these groups started to make better decisions and better thought-out choices. They were learning from their mistakes.

In every game that was played, at the end of eleven turns not one group was worse off than when it started. All groups were successful. Some, however, were more successful than others. These were the groups that carefully thought out the different possibilities and used their past mistakes and correct decisions to their advantage. □

To aid those in search of simulations suitable for schools, the following list contains several companies whose product line includes simulations.

MECC
2520 Broadway Dr.
St. Paul, MN 55113
Software includes ecology simulations for the Apple and TRS-80, and MECC software for the Apple.

Conduit
P.O. Box 388
Iowa City, IA 52244
Conduit's line includes chemistry, ecology and physics simulations for TRS-80, PET, and Apple.

Edu-Ware Services, Inc.
2222 Sherman Way, Suite 102
Canoga Park, CA 91303
Markets a line of unusual simulations for the Apple.

Atari
1196 Borregas Ave.
Sunnyvale, CA 94086
Has a variety of educational simulations such as Lemonade and Kingdom for the Atari.

Civil War

This simulation is based on 14 battles in the Civil War. Facts and figures used are based on the actual occurrence. If you follow the same strategy used in the actual battle, the results will be the same. Generally, this is a good strategy since the generals in the Civil War were fairly good military strategists. However, you can frequently outperform the Civil War generals, particularly in cases where they did not have good enemy intelligence and consequently followed a poor course of action. Naturally, it helps to know your Civil War history, although the computer gives you the rudiments.

After each of the 14 battles, your casualties are compared to the actual casualties of the battle, and you are told whether you win or lose the battle.

You may play Civil War alone in which case the program simulates the Union general. Or two players may play in which case the computer becomes the moderator.

Civil War was written in 1968 by three Students at Lexington High School, Massachusetts: L. Cram, L. Goodie, and D. Hibbard. It was modified into a 2-player game by G. Paul and R. Hess of TIES, St. Paul, Minnesota.

```
          CIVIL WAR
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


DO YOU WANT INSTRUCTIONS? YES


THIS IS A CIVIL WAR SIMULATION.
TO PLAY, TYPE A RESPONSE WHEN THE COMPUTER ASKS.
REMEMBER THAT ALL FACTORS ARE INTERRELATED AND THAT YOUR
RESPONSES COULD CHANGE HISTORY. FACTS AND FIGURES USED ARE
BASED ON THE ACTUAL OCCURRENCE. MOST BATTLES TEND TO RESULT
AS THEY DID IN THE CIVIL WAR, BUT IT ALL DEPENDS ON YOU!!

THE OBJECT OF THE GAME IS TO WIN AS MANY BATTLES AS POSSIBLE.

YOUR CHOICES FOR DEFENSIVE STRATEGY ARE:
        (1) ARTILLERY ATTACK
        (2) FORTIFICATION AGAINST FRONTAL ATTACK
        (3) FORTIFICATION AGAINST FLANKING MANEUVERS
        (4) FALLING BACK
 YOUR CHOICES FOR OFFENSIVE STRATEGY ARE:
        (1) ARTILLERY ATTACK
        (2) FRONTAL ATTACK
        (3) FLANKING MANEUVERS
        (4) ENCIRCLEMENT
YOU MAY SURRENDER BY TYPING A '5' FOR YOUR STRATEGY.
```

```
ARE THERE TWO GENERALS PRESENT (ANSWER YES OR NO)? YES
SELECT A BATTLE BY TYPING A NUMBER FROM 1 TO 14 ON
REQUEST.  TYPE ANY OTHER NUMBER TO END THE SIMULATION.
BUT '0' BRINGS BACK EXACT PREVIOUS BATTLE SITUATION
ALLOWING YOU TO REPLAY IT

NOTE: A NEGATIVE FOOD$ ENTRY CAUSES THE PROGRAM TO
USE THE ENTRIES FROM THE PREVIOUS BATTLE

AFTER REQUESTING A BATTLE, DO YOU WISH BATTLE DESCRIPTIONS
(ANSWER YES OR NO)? YES


WHICH BATTLE DO YOU WISH TO SIMULATE? 3


THIS IS THE BATTLE OF SEVEN DAYS
JUNE 25-JULY 1, 1862.  GENERAL LEE (CSA) UPHELD THE
OFFENSIVE THROUGHOUT THE BATTLE AND FORCED GEN. MCCLELLAN
AND THE UNION FORCES AWAY FROM RICHMOND.

                CONFEDERACY     UNION
MEN               95000          115000
MONEY           $ 427500       $ 517500
INFLATION         25 %           10 %

CONFEDERATE GENERAL---HOW MUCH DO YOU WISH TO SPEND FOR
 - FOOD..... ? 100000
 - SALARIES.. ? 140000
 - AMMUNITION ? 180000

UNION GENERAL---HOW MUCH DO YOU WISH TO SPEND FOR
 - FOOD..... ? 120000
 - SALARIES.. ? 160000
 - AMMUNITION ? 237500

CONFEDERATE MORALE IS FAIR
      UNION MORALE IS FAIR
CONFEDERATE GENERAL---YOU ARE ON THE OFFENSIVE

CONFEDERATE STRATEGY  ? 4
UNION STRATEGY  ? 2

                CONFEDERACY     UNION
CASUALTIES        18805          13738
DESERTIONS        13             10

COMPARED TO THE ACTUAL CASUALTIES AT SEVEN DAYS
CONFEDERATE: 91 % OF THE ORIGINAL
UNION:       87 % OF THE ORIGINAL

THE UNION WINS SEVEN DAYS
---------------
```

```
WHICH BATTLE DO YOU WISH TO SIMULATE? 6



THIS IS THE BATTLE OF FREDERICKSBURG
DEC 13, 1862.  THE CONFEDERACY UNDER LEE SUCCESSFULLY
REPULSED AN ATTACK BY THE UNION UNDER GEN. BURNSIDE.

               CONFEDERACY     UNION
MEN                 76417        122191
MONEY            $ 335800     $ 552000
INFLATION           27 %         8 %

CONFEDERATE GENERAL---HOW MUCH DO YOU WISH TO SPEND FOR
 - FOOD..... ? 100000
 - SALARIES.. ? 100000
 - AMMUNITION ? 135800

UNION GENERAL---HOW MUCH DO YOU WISH TO SPEND FOR
 - FOOD..... ? 130000
 - SALARIES.. ? 150000
 - AMMUNITION ? 272000

CONFEDERATE MORALE IS FAIR
     UNION MORALE IS HIGH
CONFEDERATE GENERAL---YOU ARE ON THE DEFENSIVE

CONFEDERATE STRATEGY  ? 2
UNION STRATEGY  ? 4

              CONFEDERACY    UNION
CASUALTIES        4870         10360
DESERTIONS          11             6

COMPARED TO THE ACTUAL CASUALTIES AT FREDERICKSBURG
CONFEDERATE: 91 % OF THE ORIGINAL
UNION:       82 % OF THE ORIGINAL

THE CONFEDERACY WINS FREDERICKSBURG
---------------



WHICH BATTLE DO YOU WISH TO SIMULATE? 15



THE CONFEDERACY HAS WON  1  BATTLES AND LOST  1
THE UNION HAS WON THE WAR

FOR THE  2  BATTLES FOUGHT (EXCLUDING RERUNS)
                              CONFEDERACY    UNION
HISTORICAL LOSSES               25991         28502
SIMULATED  LOSSES               23700         24115

    % OF ORIGINAL               91             85
```

```
2 PRINT TAB(26) "CIVIL WAR"
4 PRINT TAB(15) "CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
6 PRINT : PRINT : PRINT
20 REM ORIGINAL GAME DESIGN: CRAM, GOODIE, HIBBARD LEXINGTON H.S.
30 REM MODIFICATIONS: G. PAUL, R. HESS (TIES), 1973
50  DIM S(4),C$(14),M1(14),M2(14),C1(14),C2(14),M(14)
60  REM UNION INFO ON LIKELY CONFEDERATE STRATEGY
70  S(1)=25 : S(2)=25 : S(3)=25 : S(4)=25
82  REM READ HISTORICAL DATA.
84  FOR D=1 TO 14
86  READ C$(D),M1(D),M2(D),C1(D),C2(D),M(D)
88  NEXT D
89  LET D=RND(-1)
90  PRINT
100  PRINT "DO YOU WANT INSTRUCTIONS";
110  INPUT X$
120  IF X$="YES" THEN 160
130  IF X$="NO" THEN 370
140  PRINT "YES OR NO -- ";
150  GOTO 110
160  PRINT "
170  PRINT "THIS IS A CIVIL WAR SIMULATION."
180  PRINT "TO PLAY, TYPE A RESPONSE WHEN THE COMPUTER ASKS."
190  PRINT "REMEMBER THAT ALL FACTORS ARE INTERRELATED AND THAT YOUR"
200  PRINT "RESPONSES COULD CHANGE HISTORY. FACTS AND FIGURES USED ARE"
210  PRINT "BASED ON THE ACTUAL OCCURRENCE. MOST BATTLES TEND TO RESULT"
220  PRINT "AS THEY DID IN THE CIVIL WAR, BUT IT ALL DEPENDS ON YOU!!"
230  PRINT
240  PRINT "THE OBJECT OF THE GAME IS TO WIN AS MANY BATTLES AS ";
245  PRINT "POSSIBLE."
250  PRINT
260  PRINT "YOUR CHOICES FOR DEFENSIVE STRATEGY ARE:"
270  PRINT "         (1) ARTILLERY ATTACK"
280  PRINT "         (2) FORTIFICATION AGAINST FRONTAL ATTACK"
290  PRINT "         (3) FORTIFICATION AGAINST FLANKING MANEUVERS"
300  PRINT "         (4) FALLING BACK"
310  PRINT " YOUR CHOICES FOR OFFENSIVE STRATEGY ARE:"
320  PRINT "         (1) ARTILLERY ATTACK"
330  PRINT "         (2) FRONTAL ATTACK"
340  PRINT "         (3) FLANKING MANEUVERS"
350  PRINT "         (4) ENCIRCLEMENT"
360  PRINT "YOU MAY SURRENDER BY TYPING A '5' FOR YOUR STRATEGY."
370  PRINT "

ARE THERE TWO GENERALS PRESENT ";



380  PRINT "(ANSWER YES OR NO)";
390  INPUT B$
400  IF B$="YES" THEN 430
410  IF B$ <> "NO" THEN 380
420  PRINT "
          YOU ARE THE CONFEDERACY.   GOOD LUCK!"
425  PRINT
430  LET D=1
440  IF B$ <> "YES" THEN 460
450  LET D=2
460  PRINT "SELECT A BATTLE BY TYPING A NUMBER FROM 1 TO 14 ON"
470  PRINT "REQUEST.  TYPE ANY OTHER NUMBER TO END THE SIMULATION."
480  PRINT "BUT '0' BRINGS BACK EXACT PREVIOUS BATTLE SITUATION"
490  PRINT "ALLOWING YOU TO REPLAY IT"
500  PRINT
510  PRINT "NOTE: A NEGATIVE FOOD$ ENTRY CAUSES THE PROGRAM TO "
520  PRINT "USE THE ENTRIES FROM THE PREVIOUS BATTLE"
530  PRINT
540  PRINT "AFTER REQUESTING A BATTLE, DO YOU WISH ";
550  PRINT "BATTLE DESCRIPTIONS ";
560  PRINT "(ANSWER YES OR NO)";
570  INPUT X$
580  IF X$="YES" THEN 600
590  IF X$ <> "NO" THEN 560
600  L=0:W=0:R1=0:Q1=0:N3=0:N4=0:P1=0:P2=0:T1=0:T2=0
610  F(2)=0:H(2)=0:B(2)=0:R2=0:Q2=0:C6=0:F=0:W0=0:Y=0:Y2=0:U=0:U2=0
620  PRINT "



630  PRINT "WHICH BATTLE DO YOU WISH TO SIMULATE";
640  INPUT A
650  IF A <> 0 THEN 660
655  IF R <> 0 THEN 1140
660  IF A <= 0 THEN 2860
665  IF A >= 15 THEN 2860
670  LET C$=C$(A)
680  LET M1=M1(A)
690  LET M2=M2(A)
700  LET C1=C1(A)
710  LET C2=C2(A)
720  LET M=M(A)
960  LET U=0
970  REM  INFLATION CALC
980  LET I1=10+(L-W)*2
990  LET I2=10+(W-L)*2
1000  REM - MONEY AVAILABLE
1010  LET B(1)=100*INT((M1*(100-I1)/2000)*(1+(R1-Q1)/(R1+1))+.5)
1020  LET B(2)=100*INT(M2*(100-I2)/2000+.5)
1030  IF B$ <> "YES" THEN 1050
1040  LET B(2)=100*INT((M2*(100-I2)/2000)*(1+(R2-Q2)/(R2+1))+.5)
1050  REM - MEN   AVAILABLE
1060  LET N5=INT(M1*(1+(P1-T1)/(N3+1)))
1070  LET N6=INT(M2*(1+(P2-T2)/(N4+1)))
1080  LET F1=5*M1/6
1090  PRINT "
```

```
          "
1100  PRINT "THIS IS THE BATTLE OF ";C$
1110  IF X$="NO" THEN 1150
1120  IF A>11 THEN 1130
1125  ON A GOTO 3580,3620,3650,3690,3720,3750,3780,3800,3830,3860,3890
1130  ON A-11 GOTO 3920,3950,3980
1140  PRINT C$" INSTANT REPLAY"
1150  PRINT
1160  PRINT " ","CONFEDERACY"," UNION"
1170  PRINT "MEN"," M5," M6
1180  PRINT "MONEY","$";B(1),"$";B(2)
1190  PRINT "INFLATION"," ";I1+15;"%"," ";I2;"%"
1195  PRINT
1200  REM - ONLY IN PRINTOUT IS CONFED INFLATION = I1+15%
1210  REM - IF TWO GENERALS, INPUT CONFED. FIRST
1220  FOR I=1 TO D
1230  IF B$ <> "YES" THEN 1260
1240  IF I=2 THEN 1260
1250  PRINT "CONFEDERATE GENERAL---";
1260  PRINT "HOW MUCH DO YOU WISH TO SPEND FOR"
1270  PRINT " - FOOD.....";
1280  INPUT F
1290  IF F >= 0 THEN 1360
1300  IF R1 <> 0 THEN 1330
1310  PRINT "NO PREVIOUS ENTRIES"
1320  GOTO 1270
1330  PRINT "ASSUME YOU WANT TO KEEP SAME ALLOCATONS"
1340  PRINT
1350  GOTO 1510
1360  LET F(I)=F
1370  PRINT " - SALARIES..";
1380  INPUT H(I)
1390  LET N=1
1400  IF H(I)<0 THEN 1490
1410  PRINT " - AMMUNITION";
1420  INPUT B(I)
1430  LET N=2
1440  IF B(I)<0 THEN 1490
1450  PRINT
1460  IF F(I)+H(I)+B(I) <= B(I) THEN 1510
1470  PRINT "THINK AGAIN! YOU HAVE ONLY $"B(I)
1480  GOTO 1270
1490  PRINT "NEGATIVE VALUES NOT ALLOWED."
1500  ON N GOTO 1370,1410
1510  IF B$ <> "YES" THEN 1550
1520  IF I=2 THEN 1550
1530  PRINT "UNION GENERAL---";
1540  NEXT I
1550  FOR Z=1 TO D
1560  IF B$ <> "YES" THEN 1620
1570  ON Z GOTO 1580,1600
1580  PRINT "CONFEDERATE ";
1590  GOTO 1620
1600  PRINT "      UNION ";
1610  REM - FIND MORALE
1620  LET O=((2*F(Z)^2+H(Z)^2)/F1^2+1)
1630  IF O<10 THEN 1660
1640  PRINT "MORALE IS HIGH"
1650  GOTO 1700
1660  IF O<5 THEN 1690
1670  PRINT "MORALE IS FAIR"
1680  GOTO 1700
1690  PRINT "MORALE IS POOR"
1700  IF B$ <> "YES" THEN 1760
1710  LET O(Z)=O
1720  NEXT Z
1730  LET O2=O(2)
1740  LET O=O(1)
1750  PRINT "CONFEDERATE GENERAL---";
1760  REM - ACTUAL OFF/DEF BATTLE SITUATION
1770  IF N <> 3 THEN 1800
1780  PRINT "YOU ARE ON THE OFFENSIVE"
1790  GOTO 1840
1800  IF N <> 1 THEN 1830
1810  PRINT "YOU ARE ON THE DEFENSIVE"
1820  GOTO 1840
1830  PRINT "BOTH SIDES ARE ON THE OFFENSIVE "
1840  PRINT
1850  REM - CHOOSE STRATEGIES
1860  IF B$ <> "YES" THEN 1910
1870  FOR I=1 TO 2
1880  ON I GOTO 1890,1920
1890  PRINT "CONFEDERATE STRATEGY ";
1900  GOTO 1920
1910  PRINT "YOUR STRATEGY ";
1920  INPUT Y
1930  IF ABS(Y-3)<3 THEN 1960
1940  PRINT "STRATEGY";Y;"NOT ALLOWED."
1950  GOTO 1910
1960  IF B$="YES" THEN 2000
1970  IF Y=5 THEN 2830
1980  GOSUB 3110
1990  GOTO 2170
2000  IF I=2 THEN 2040
2010  LET Y1=Y
2020  PRINT "UNION STRATEGY ";
2030  NEXT I
2040  LET Y2=Y
2050  LET Y=Y1
2060  IF Y2=5 THEN 2020
2070  REM : SIMULATED LOSSES-NORTH
2080  LET C6=(2*C2/5)*(1+1/(2*(ABS(Y2-Y)+1)))
2090  LET C6=C6*(1.28+(5*M2/6)/(B(2)+1))
2100  LET C6=INT(C6*(1+1/02)+.5)
2110  REM - IF LOSS > MEN PRESENT, RESCALE LOSSES
2120  LET E2=100/02
2130  IF INT(C6+E2)<M6 THEN 2190
2140  LET C6=INT(13*M6/20)
2150  LET E2=7*C6/13
2160  LET U2=1
2170  REM - CALCULATE SIMULATED LOSSES
2180  PRINT
2190  PRINT "

            ","CONFEDERACY","UNION"
2200  LET C5=(2*C1/5)*(1+1/(2*(ABS(Y2-Y)+1)))
2210  LET C5=INT(C5*(1+1/D)*(1.28+F1/(B(1)+1)))+.5)
2220  LET E=100/D
2230  IF C5+100/D<M1*(1+(P1-T1)/(M3+1)) THEN 2270
2240  LET C5=INT(13*M1/20*(1+(P1-T1)/(M3+1)))
2250  LET E=7*C5/13
2260  LET U=1
2270  IF D=1 THEN 2500
2280  PRINT "CASUALTIES",C5,C6
2290  PRINT "DESERTIONS",INT(E),INT(E2)
2300  PRINT
2310  IF B$ <> "YES" THEN 2530
2320  PRINT "COMPARED TO THE ACTUAL CASUALTIES AT "C$
2330  PRINT "CONFEDERATE:"INT(100*(C5/C1)+.5)"% OF THE ORIGINAL"
2340  PRINT "UNION:      "INT(100*(C6/C2)+.5)"% OF THE ORIGINAL"
2350  PRINT
2360  REM - 1 WHO ONE
2370  IF U <> 1 THEN 2380
2375  IF U2=1 THEN 2460
2380  IF U=1 THEN 2420
2390  IF U2=1 THEN 2440
2400  IF C5+E=C6+E2 THEN 2460
2410  IF C5+E<C6+E2 THEN 2440
2420  PRINT "THE UNION WINS "C$
2430  GOTO 2600
2440  PRINT "THE CONFEDERACY WINS "C$
2450  GOTO 2660
2460  PRINT "BATTLE OUTCOME UNRESOLVED"
2470  LET W0=W0+1
2480  IF A=0 THEN 2790
2490  GOTO 2680
2500  LET C6=INT(17*C2*C1/(C5*20))
2510  LET E2=5*0
2520  GOTO 2280
2530  PRINT "YOUR CASUALTIES WERE "INT(100*(C5/C1)+.5)"% OF "
2540  PRINT "THE ACTUAL CASUALTIES AT ";C$
2550  PRINT
2560  REM - FIND WHO WON
2570  IF U=1 THEN 2590
2580  IF C5+E<17*C2*C1/(C5*20)+5*0 THEN 2630
2590  PRINT "YOU LOSE ";C$
2600  IF A=0 THEN 2790
2610  LET L=L+1
2620  GOTO 2680
2630  PRINT "YOU WIN ";C$
2640  REM - CUMULATIVE BATTLE FACTORS WHICH ALTER HISTORICAL
2650  REM   RESOURCES AVAILABLE.IF A REPLAY DON'T UPDATE.
2660  IF A=0 THEN 2790
2670  LET W=W+1
2680  LET T1=T1+C5+E
2690  LET T2=T2+C6+E2
2700  LET P1=P1+C1
2710  LET P2=P2+C2
2720  LET Q1=Q1+(F(1)+H(1)+B(1))
2730  LET Q2=Q2+(F(2)+H(2)+B(2))
2740  LET R1=R1+M1*(100-I1)/20
2750  LET R2=R2+M2*(100-I2)/20
2760  LET M3=M3+M1
2770  LET M4=M4+M2
2780  GOSUB 3300
```

```
2790  U=0:U2=0
2800  PRINT "---------------"
2810  GOTO 620
2820  REM------FINISH OFF
2830  PRINT "THE CONFEDERACY HAS SURRENDERED"
2840  GOTO 2860
2850  PRINT "THE UNION HAS SURRENDERED."
2860  PRINT "
            "
2870  PRINT "THE CONFEDERACY ";
2880  PRINT "HAS WON "W" BATTLES AND LOST "L
2890  IF Y=5 THEN 2940
2900  IF Y2=5 THEN 2920
2910  IF W <= L THEN 2940
2915  IF Y=5 THEN 2940
2920  PRINT "THE CONFEDERACY HAS WON THE WAR"
2930  GOTO 2950
2940  PRINT "THE UNION HAS WON THE WAR"
2950  PRINT
2960  IF R1=0 THEN 3100
2970  PRINT "FOR THE "W+L+WO" BATTLES FOUGHT (EXCLUDING RERUNS)"
2980  PRINT " "," "," ";
2990  PRINT "CONFEDERACY"," UNION"
3000  PRINT "HISTORICAL LOSSES",INT(P1+.5),INT(P2+.5)
3010  PRINT "SIMULATED  LOSSES",INT(T1+.5),INT(T2+.5)
3020  PRINT
3030  PRINT "    % OF ORIGINAL",INT(100*(T1/P1)+.5),INT(100*(T2/P2)+.5)
3040  IF B$="YES" THEN 3100
3050  PRINT
3060  PRINT "UNION INTELLIGENCE SUGGESTS THAT THE SOUTH USED "
3070  PRINT "STRATEGIES 1, 2, 3, 4 IN THE FOLLOWING PERCENTAGES"
3080  PRINT S(1);S(2);S(3);S(4)
3090  REM---------------------------------
3100  STOP
3110  REM - UNION STRATEGY IS COMPUTER CHOSEN
3120  PRINT "UNION STRATEGY IS ";
3130  IF A <> 0 THEN 3180
3140  INPUT Y2
3150  IF Y2 <= 0 THEN 3160
3155  IF Y2<5 THEN 3290
3160  PRINT "ENTER 1 , 2 ,3 , OR 4 (USUALLY PREVIOUS UNION STRATEGY)"
3170  GOTO 3140
3180  LET S0=0
3190  LET R=100*RND(0)
3200  FOR I=1 TO 4
3210  LET S0=S0+S(I)
3220  REM - IF ACTUAL STRATEGY INFO IS IN PROGRAM DATA STATEMENTS
3230  REM   THEN R-100 IS EXTRA WEIGHT GIVEN TO THAT STATEGY.
3240  IF R<S0 THEN 3270
3250  NEXT I
3260  REM - IF ACTUAL STRAT. IN,THEN HERE IS Y2= HIST. STRAT.
3270  LET Y2=I
3280  PRINT Y2
3290  RETURN
3300  REM LEARN  PRESENT STRATEGY, START FORGETTING OLD ONES
3310  REM - PRESENT STRATEGY OF SOUTH GAINS 3*S, OTHERS LOSE S
3320  REM   PROBABILITY POINTS, UNLESS A STRATEGY FALLS BELOW 5%.
3330  LET S=3
3340  LET S0=0
3350  FOR I=1 TO 4
3360  IF S(I) <= 5 THEN 3390
3370  LET S(I)=S(I)-S
3380  LET S0=S0+S
3390  NEXT I
3400  LET S(Y)=S(Y)+S0
3410  RETURN
3420  REM - HISTORICAL DATA...CAN ADD MORE (STRAT.,ETC) BY INSERTING
3430  REM   DATA STATEMENTS AFTER APPRO. INFO, AND ADJUSTING READ
3440  DATA "BULL RUN",18000,18500,1967,2708,1
3450  DATA "SHILOH",40000.,44894.,10699,13047,3
3460  DATA "SEVEN DAYS",95000.,115000.,20614,15849,3
3470  DATA "SECOND BULL RUN",54000.,63000.,10000,14000,2
3480  DATA "ANTIETAM",40000.,50000.,10000,12000,3
3490  DATA "FREDERICKSBURG",75000.,120000.,5377,12653,1
3500  DATA "MURFREESBORO",38000.,45000.,11000,12000,1
3510  DATA "CHANCELLORSVILLE",32000,90000.,13000,17197,2
3520  DATA "VICKSBURG",50000.,70000.,12000,19000,1
3530  DATA "GETTYSBURG",72500.,85000.,20000,23000,3
3540  DATA "CHICKAMAUGA",66000.,60000.,18000,16000,2
3550  DATA "CHATTANOOGA",37000.,60000.,36700.,5800,2
3560  DATA "SPOTSYLVANIA",62000.,110000.,17723,18000,2
3570  DATA "ATLANTA",65000.,100000.,8500,3700,1
3580  PRINT "JULY 21, 1861.  GEN. BEAUREGARD, COMMANDING THE SOUTH, MET"
3590  PRINT "UNION FORCES WITH GEN. MCDOWELL IN A PREMATURE BATTLE AT"
3600  PRINT "BULL RUN.  GEN. JACKSON HELPED PUSH BACK THE UNION ATTACK."
3610  GOTO 1150
3620  PRINT "APRIL 6-7, 1862.  THE CONFEDERATE SURPRISE ATTACK AT"
3630  PRINT "SHILOH FAILED DUE TO POOR ORGANIZATION."
3640  GOTO 1150
3650  PRINT "JUNE 25-JULY 1, 1862.  GENERAL LEE (CSA) UPHELD THE"
3660  PRINT "OFFENSIVE THROUGHOUT THE BATTLE AND FORCED GEN. MCCLELLAN"
3670  PRINT "AND THE UNION FORCES AWAY FROM RICHMOND."
3680  GOTO 1150
3690  PRINT "AUG 29-30, 1862.  THE COMBINED CONFEDERATE FORCES UNDER";
3695  PRINT " LEE"
3700  PRINT "AND JACKSON DROVE THE UNION FORCES BACK INTO WASHINGTON."
3710  GOTO 1150
3720  PRINT "SEPT 17, 1862.  THE SOUTH FAILED TO INCORPORATE MARYLAND"
3730  PRINT "INTO THE CONFEDERACY."
3740  GOTO 1150
3750  PRINT "DEC 13, 1862.  THE CONFEDERACY UNDER LEE SUCCESSFULLY"
3760  PRINT "REPULSED AN ATTACK BY THE UNION UNDER GEN. BURNSIDE."
3770  GOTO 1150
3780  PRINT "DEC 31, 1862.  THE SOUTH UNDER GEN. BRAGG WON A CLOSE ";
3785  PRINT "BATTLE."
3790  GOTO 1150
3800  PRINT "MAY 1-6, 1863.  THE SOUTH HAD A COSTLY VICTORY AND LOST"
3810  PRINT "ONE OF THEIR OUTSTANDING GENERALS, 'STONEWALL' JACKSON."
3820  GOTO 1150
3830  PRINT "JULY 4, 1863.  VICKSBURG WAS A COSTLY DEFEAT FOR THE SOUTH"
3840  PRINT "BECAUSE IT GAVE THE UNION ACCESS TO THE MISSISSIPPI."
3850  GOTO 1150
3860  PRINT "JULY 1-3, 1863.  A SOUTHERN MISTAKE BY GEN. LEE AT ";
3865  PRINT "GETTYSBURG"
3870  PRINT "COST THEM ONE OF THE MOST CRUCIAL BATTLES OF THE WAR."
3880  GOTO 1150
3890  PRINT "SEPT. 15, 1863. CONFUSION IN A FOREST NEAR CHICKAMAUGA LED"
3900  PRINT "TO A COSTLY SOUTHERN VICTORY."
3910  GOTO 1150
3920  PRINT "NOV. 25, 1863. AFTER THE SOUTH HAD SIEGED GEN. ROSENCRANS'"
3930  PRINT "ARMY FOR THREE MONTHS, GEN. GRANT BROKE THE SIEGE."
3940  GOTO 1150
3950  PRINT "MAY 5, 1864.  GRANT'S PLAN TO KEEP LEE ISOLATED BEGAN TO"
3960  PRINT "FAIL HERE, AND CONTINUED AT COLD HARBOR AND PETERSBURG."
3970  GOTO 1150
3980  PRINT "AUGUST, 1864.  SHERMAN AND THREE VETERAN ARMIES CONVERGED"
3990  PRINT "ON ATLANTA AND DEALT THE DEATH BLOW TO THE CONFEDERACY."
4000  GOTO 1150
4010  END
```

Fur Trader

You are the leader of a French fur trading expedition in 1776 leaving the Ontario area to sell furs and get supplies for the next year. You have a choice of three forts at which you may trade. The cost of supplies and the amount you receive for your furs will depend upon the fort you choose. You also specify what types of furs that you have to trade.

The game goes on and on until you elect to trade no longer.

Author of the program is Dan Bachor, University of Calgary, Alberta, Canada.

```
                  FUR TRADER
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

YOU ARE THE LEADER OF A FRENCH FUR TRADING EXPEDITION IN
1776 LEAVING THE LAKE ONTARIO AREA TO SELL FURS AND GET
SUPPLIES FOR THE NEXT YEAR.  YOU HAVE A CHOICE OF THREE
FORTS AT WHICH YOU MAY TRADE.  THE COST OF SUPPLIES
AND THE AMOUNT YOU RECEIVE FOR YOUR FURS WILL DEPEND
ON THE FORT THAT YOU CHOOSE.
DO YOU WISH TO TRADE FURS?
ANSWER YES OR NO            ? YES

YOU HAVE $ 600 SAVINGS.
AND 190 FURS TO BEGIN THE EXPEDITION.

YOUR 190 FURS ARE DISTRIBUTED AMONG THE FOLLOWING
KINDS OF PELTS: MINK, BEAVER, ERMINE AND FOX.

HOW MANY MINK PELTS DO YOU HAVE? 40

HOW MANY BEAVER PELTS DO YOU HAVE? 50

HOW MANY ERMINE PELTS DO YOU HAVE? 60

HOW MANY FOX PELTS DO YOU HAVE? 40
DO YOU WANT TO TRADE YOUR FURS AT FORT 1, FORT 2,
OR FORT 3?  FORT 1 IS FORT HOCHELAGA (MONTREAL)
AND IS UNDER THE PROTECTION OF THE FRENCH ARMY.
FORT 2 IS FORT STADACONA (QUEBEC) AND IS UNDER THE
PROTECTION OF THE FRENCH ARMY.  HOWEVER, YOU MUST
MAKE A PORTAGE AND CROSS THE LACHINE RAPIDS.
FORT 3 IS FORT NEW YORK AND IS UNDER DUTCH CONTROL.
YOU MUST CROSS THROUGH IROQUOIS LAND.
ANSWER 1, 2, OR 3.
? 3
YOU HAVE CHOSEN THE MOST DIFFICULT ROUTE.  AT
FORT NEW YORK YOU WILL RECEIVE THE HIGHEST VALUE
FOR YOUR FURS.  THE COST OF YOUR SUPPLIES
WILL BE LOWER THAN AT ALL THE OTHER FORTS.
DO YOU WANT TO TRADE AT ANOTHER FORT?
ANSWER YES OR NO            ? YES
ANSWER 1, 2, OR 3.
? 2
YOU HAVE CHOSEN A HARD ROUTE.  IT IS, IN COMPARSION,
HARDER THAN THE ROUTE TO HOCHELAGA BUT EASIER THAN
THE ROUTE TO NEW YORK.  YOU WILL RECEIVE AN AVERAGE VALUE
FOR YOUR FURS AND THE COST OF YOUR SUPPLIES WILL BE AVERAGE
DO YOU WANT TO TRADE AT ANOTHER FORT?
ANSWER YES OR NO            ? NO

YOUR CANOE UPSET IN THE LACHINE RAPIDS.  YOU
LOST ALL YOUR FURS
SUPPLIES AT FORT STADACONA COST $125.00
YOUR TRAVEL EXPENSES TO STADACONA WERE $15.00

YOU NOW HAVE $ 460  INCLUDING YOUR PREVIOUS SAVINGS

DO YOU WANT TO TRADE FURS NEXT YEAR?
ANSWER YES OR NO            ? YES

YOU HAVE $ 460 SAVINGS.
AND 190 FURS TO BEGIN THE EXPEDITION.

YOUR 190 FURS ARE DISTRIBUTED AMONG THE FOLLOWING
KINDS OF PELTS: MINK, BEAVER, ERMINE AND FOX.

HOW MANY MINK PELTS DO YOU HAVE? 50

HOW MANY BEAVER PELTS DO YOU HAVE? 100

HOW MANY ERMINE PELTS DO YOU HAVE? 20

HOW MANY FOX PELTS DO YOU HAVE? 20
DO YOU WANT TO TRADE YOUR FURS AT FORT 1, FORT 2,
OR FORT 3?  FORT 1 IS FORT HOCHELAGA (MONTREAL)
AND IS UNDER THE PROTECTION OF THE FRENCH ARMY.
FORT 2 IS FORT STADACONA (QUEBEC) AND IS UNDER THE
PROTECTION OF THE FRENCH ARMY.  HOWEVER, YOU MUST
MAKE A PORTAGE AND CROSS THE LACHINE RAPIDS.
FORT 3 IS FORT NEW YORK AND IS UNDER DUTCH CONTROL.
YOU MUST CROSS THROUGH IROQUOIS LAND.
ANSWER 1, 2, OR 3.
? 3
YOU HAVE CHOSEN THE MOST DIFFICULT ROUTE.  AT
FORT NEW YORK YOU WILL RECEIVE THE HIGHEST VALUE
FOR YOUR FURS.  THE COST OF YOUR SUPPLIES
WILL BE LOWER THAN AT ALL THE OTHER FORTS.
DO YOU WANT TO TRADE AT ANOTHER FORT?
ANSWER YES OR NO            ? NO

YOU NARROWLY ESCAPED AN IROQUOIS RAIDING PARTY.
HOWEVER, YOU HAD TO LEAVE ALL YOUR FURS BEHIND.
SUPPLIES AT NEW YORK COST $80.00
YOUR TRAVEL EXPENSES TO NEW YORK WERE $25.00

YOU NOW HAVE $ 355  INCLUDING YOUR PREVIOUS SAVINGS

DO YOU WANT TO TRADE FURS NEXT YEAR?
ANSWER YES OR NO            ? NO
```

```
1 DIM F(4)
2 PRINT TAB(31);"FUR TRADER"
4 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
6 PRINT: PRINT: PRINT
15 GOSUB 1091
16 LET I=600
17 PRINT "DO YOU WISH TO TRADE FURS?"
18 GOSUB 1402
19 IF B$="YES" THEN 100
20 IF B$="YES " THEN 100
21 STOP
100 PRINT
101 PRINT "YOU HAVE $";I "SAVINGS."
102 PRINT "AND 190 FURS TO BEGIN THE EXPEDITION."
261 LET E1=INT((.15*RND(1)+.95)*10^2+.5)/10^2
262 LET B1=INT((.25*RND(1)+1.00)*10^2+.5)/10^2
300 PRINT
301 PRINT "YOUR 190 FURS ARE DISTRIBUTED AMONG THE FOLLOWING"
302 PRINT "KINDS OF PELTS: MINK, BEAVER, ERMINE AND FOX."
310 GOSUB 1430
315 RESTORE
330 FOR J=1 TO 4
332 READ B$
333 PRINT
335 PRINT "HOW MANY ";B$;" PELTS DO YOU HAVE";
338 INPUT F(J)
340 LET F(0)=F(1)+F(2)+F(3)+F(4)
342 IF F(0)=190 THEN 1100
344 IF F(0)>190 THEN 500
348 NEXT J
350 GOTO 1100
500 PRINT
501 PRINT "YOU MAY NOT HAVE THAT MANY FURS."
502 PRINT "DO NOT TRY TO CHEAT.  I CAN ADD."
503 PRINT "YOU MUST START AGAIN."
504 GOTO 15
508 PRINT
511 PRINT "DO YOU WANT TO TRADE FURS NEXT YEAR?"
513 GOTO 18
1091 PRINT "YOU ARE THE LEADER OF A FRENCH FUR TRADING EXPEDITION IN "
1092 PRINT "1776 LEAVING THE LAKE ONTARIO AREA TO SELL FURS AND GET"
1093 PRINT "SUPPLIES FOR THE NEXT YEAR.  YOU HAVE A CHOICE OF THREE"
1094 PRINT "FORTS AT WHICH YOU MAY TRADE.  THE COST OF SUPPLIES"
1095 PRINT "AND THE AMOUNT YOU RECEIVE FOR YOUR FURS WILL DEPEND"
1096 PRINT "ON THE FORT THAT YOU CHOOSE."
1099 RETURN
1100 PRINT "DO YOU WANT TO TRADE YOUR FURS AT FORT 1, FORT 2,"
1102 PRINT "OR FORT 3?  FORT 1 IS FORT HOCHELAGA (MONTREAL)"
1103 PRINT "AND IS UNDER THE PROTECTION OF THE FRENCH ARMY."
1104 PRINT "FORT 2 IS FORT STADACONA (QUEBEC) AND IS UNDER THE"
1105 PRINT "PROTECTION OF THE FRENCH ARMY.  HOWEVER, YOU MUST"
1106 PRINT "MAKE A PORTAGE AND CROSS THE LACHINE RAPIDS."
1108 PRINT "FORT 3 IS FORT NEW YORK AND IS UNDER DUTCH CONTROL."
1109 PRINT "YOU MUST CROSS THROUGH IROQUOIS LAND."
1110 PRINT "ANSWER 1, 2, OR 3."
1111 INPUT B
1112 IF B=1 THEN 1120
1113 IF B=2 THEN 1135
1115 IF B=3 THEN 1147
1116 GOTO 1110
1120 PRINT "YOU HAVE CHOSEN THE EASIEST ROUTE.  HOWEVER, THE FORT"
1121 PRINT "IS FAR FROM ANY SEAPORT.  THE VALUE"
1122 PRINT "YOU RECEIVE FOR YOUR FURS WILL BE LOW AND THE COST"
1123 PRINT "OF SUPPLIES HIGHER THAN AT FORTS STADACONA OR NEW YORK."
1125 GOSUB 1400
1129 IF B$="YES" THEN 1110
1130 GOTO 1160
1135 PRINT "YOU HAVE CHOSEN A HARD ROUTE.  IT IS, IN COMPARISON,"
1136 PRINT "HARDER THAN THE ROUTE TO HOCHELAGA BUT EASIER THAN"
1137 PRINT "THE ROUTE TO NEW YORK.  YOU WILL RECEIVE AN AVERAGE VALUE"
1138 PRINT "FOR YOUR FURS AND THE COST OF YOUR SUPPLIES WILL BE AVERAGE
1141 GOSUB 1400
1144 IF B$="YES" THEN 1110
1145 GOTO 1198
1147 PRINT "YOU HAVE CHOSEN THE MOST DIFFICULT ROUTE.  AT"
1148 PRINT "FORT NEW YORK YOU WILL RECEIVE THE HIGHEST VALUE"
1149 PRINT "FOR YOUR FURS.  THE COST OF YOUR SUPPLIES"
1150 PRINT "WILL BE LOWER THAN AT ALL THE OTHER FORTS."
1152 GOSUB 1400
1155 IF B$="YES" THEN 1110
1156 GOTO 1250
1160 LET I=I-160
1169 PRINT
1174 LET M1=INT((.2*RND(1)+.70)*10^2+.5)/10^2
1175 LET E1=INT((.2*RND(1)+.65)*10^2+.5)/10^2
1176 LET B1=INT((.2*RND(1)+.75)*10^2+.5)/10^2
1177 LET D1=INT((.2*RND(1)+.80)*10^2+.5)/10^2
1180 PRINT "SUPPLIES AT FORT HOCHELAGA COST $150.00"
1181 PRINT "YOUR TRAVEL EXPENSES TO HOCHELAGA WERE $10.00"
1190 GOTO 1410
1198 LET I=I-140
1201 PRINT
1205 LET M1=INT((.3*RND(1)+.85)*10^2+.5)/10^2
1206 LET E1=INT((.15*RND(1)+.80)*10^2+.5)/10^2
1207 LET B1=INT((.2*RND(1)+.90)*10^2+.5)/10^2
1209 LET P=INT(10*RND(1))+1
1210 IF P<=2 THEN 1216
1212 IF P<=6 THEN 1224
1213 IF P<=8 THEN 1226
1215 IF P<=10 THEN 1235
1216 LET F(2)=0
1218 PRINT "YOUR BEAVER WERE TOO HEAVY TO CARRY ACROSS"
1219 PRINT "THE PORTAGE.  YOU HAD TO LEAVE THE PELTS BUT FOUND"
1220 PRINT "THEM STOLEN WHEN YOU RETURNED"
1221 GOSUB 1244
1222 GOTO 1414
1224 PRINT "YOU ARRIVED SAFELY AT FORT STADACONA"
1225 GOTO 1239
1226 GOSUB 1430
1230 PRINT "YOUR CANOE UPSET IN THE LACHINE RAPIDS.  YOU"
1231 PRINT "LOST ALL YOUR FURS"
1232 GOSUB 1244
1233 GOTO 1418
1235 LET F(4)=0
1237 PRINT "YOUR FOX PELTS WERE NOT CURED PROPERLY."
1238 PRINT "NO ONE WILL BUY THEM."
1239 GOSUB 1244
1240 GOTO 1410
1244 PRINT "SUPPLIES AT FORT STADACONA COST $125.00"
1246 PRINT "YOUR TRAVEL EXPENSES TO STADACONA WERE $15.00"
1248 RETURN
1250 LET I=I-105
1254 PRINT
1260 LET M1=INT((.15*RND(1)+1.05)*10^2+.5)/10^2
1263 LET D1=INT((.25*RND(1)+1.10)*10^2+.5)/10^2
1270 LET P=INT(10*RND(1))+1
1271 IF P<=2 THEN 1281
1272 IF P<=6 THEN 1291
1273 IF P<=8 THEN 1295
1274 IF P<=10 THEN 1306
1281 PRINT "YOU WERE ATTACKED BY A PARTY OF IROQUOIS."
1282 PRINT "ALL PEOPLE IN YOUR TRADING GROUP WERE"
1283 PRINT "KILLED.  THIS ENDS THE GAME."
1284 STOP
1291 PRINT "YOU WERE LUCKY.  YOU ARRIVED SAFELY"
1292 PRINT "AT FORT NEW YORK."
1293 GOTO 1311
1295 GOSUB 1430
1300 PRINT "YOU NARROWLY ESCAPED AN IROQUOIS RAIDING PARTY."
1301 PRINT "HOWEVER, YOU HAD TO LEAVE ALL YOUR FURS BEHIND."
1303 GOSUB 1320
1304 GOTO 1418
1306 LET B1=B1/2
1307 LET M1=M1/2
1308 PRINT "YOUR MINK AND BEAVER WERE DAMAGED ON YOUR TRIP."
1309 PRINT "YOU RECEIVE ONLY HALF THE CURRENT PRICE FOR THESE FURS."
1311 GOSUB 1320
1312 GOTO 1410
1320 PRINT "SUPPLIES AT NEW YORK COST $80.00"
1321 PRINT "YOUR TRAVEL EXPENSES TO NEW YORK WERE $25.00"
1322 RETURN
1400 PRINT "DO YOU WANT TO TRADE AT ANOTHER FORT?"
1402 PRINT "ANSWER YES OR NO",
1403 INPUT B$
1404 RETURN
1410 PRINT
1412 PRINT "YOUR BEAVER SOLD FOR $";B1*F(2);
1414 PRINT "YOUR FOX SOLD FOR $";D1*F(4)
1416 PRINT "YOUR ERMINE SOLD FOR $";E1*F(3);
1417 PRINT "YOUR MINK SOLD FOR $";M1*F(1)
1418 LET I=M1*F(1)+B1*F(2)+E1*F(3)+D1*F(4)+I
1420 PRINT
1422 PRINT "YOU NOW HAVE $";I;" INCLUDING YOUR PREVIOUS SAVINGS"
1425 GOTO 508
1430 FOR J=1 TO 4
1432 LET F(J)=0
1434 NEXT J
1436 RETURN
2000 DATA "MINK","BEAVER","ERMINE","FOX"
2046 END
```

Oregon Trail

Dan Rawitsch

This program simulates a trip over the Oregon Trail from Independence, Missouri to Oregon City, Oregon in 1847. Your family of five will cover the 2040-mile Oregon Trail in 5-6 months — if you make it alive.

INTRODUCTION

"The Santa Fe Trail being first established, a signboard was later set up to show where the Oregon Trail branched off. It bore the simple legend 'Road To Oregon.' ... Surely so unostentatious a sign never before nor since announced so long a journey."

- H.M. Chittenden
The American Fur Trade of the Far West

During the thirty-year period from 1840 to 1870, thousands of pioneers traveled over the 2000-mile Oregon Trail to settle on the West Coast. The history of the trail may be seen by some as a strong example of heroic American themes such as "conquering the frontier" and "the pioneer spirit." To others, the great western migration carries the political overtones of the colonists and their descendents forcing out British imperialism and clearing away the native American Indians in an effort to dominate middle North America themselves. At the very least, the journey over the trail represents the human stories of many individuals who, oblivious to historical trends, tried to survive in life as best they knew how.

RATIONALE FOR COMPUTER USAGE

This computer simulation, developed initially in 1971 and refined in 1975, is an attempt to give students a better feeling of what the journey west was like for the people who attempted it. Like all simulations, OREGON does not attempt to replicate exactly a trip on a wagon train in the 1840's. But it does attempt to present students with some of the resources, decisions, and events that faced the pioneers of that day. Although students can find out about the Oregon Trail by reading books, visiting museums, watching movies, and similar activities, the simulation allows them to learn from actively participating in the simulated experiences of people from another era.

HISTORICAL BACKUP INFORMATION

Although historical information about the trip to Oregon is not extremely plentiful, primary and secondary sources were used whenever possible to make the simulation authentic.

Background On The OREGON Program

In 1971, Don Rawitsch and Bill Heinemann were participating together in a practice teaching program as students at Carleton College, Northfield, Minnesota. Don was teaching a class on the history of the American West and provided the preliminary information which Bill, a math teacher, used to construct the OREGON program. The program was first implemented on the Minneapolis Schools timesharing system. On the completion of the practice teaching program, the program was removed from the Minneapolis system and remained only as a curled up listing until Don joined the MECC staff in 1974 and loaded it onto the MECC system. Don then proceeded to do further research on the Oregon Trail and modified the program for historical accuracy to produce the present version. The program has been implemented on Hewlett-Packard, UNIVAC, and Control Data systems.

- Mileage and route of the Trail

  Morgan, insert-back cover, "Map of T.H. Jefferson - 1849"

  Hancock, xiv, xv, information based on a map of 1846

  Meeker, *Ox Team Days*, pp. 252-53

  Meeker, *Ox Team Days*, p. 61 - says travelers averaged 15-25 miles per day, though they didn't travel every day. In the simulation, players make about 175-200 miles every two-week period.

  Ghent, p. 73 - says ox-drawn wagons made 2 miles per hour, or 20 miles on good days and 5-10 miles on bad days.

- Costs of resources

  Meeker, *Ox Team Days*, p. 13, says in 1850's sugar cost 18¢/lb., salt cost $3.00/barrel, calico cost 15¢/yd.

  Ghent, p. 99, says a team of oxen cost about $200 (for eight); references a guidebook of the time which recommends the following to be included for each adult:
  150 lbs. of flour 25 lbs. of bacon
  25 lbs. of sugar 15 lbs. of coffee
  In the simulation, the player spends $200-$300 on an oxen team. Based on the Meeker information, if the average commodity cost about 20¢/lb. and the average family of five eats as much as four adults, a good food stock would cost about $175. This is a reasonable amount to start with in the simulation.

- Frequency of misfortunes occurring

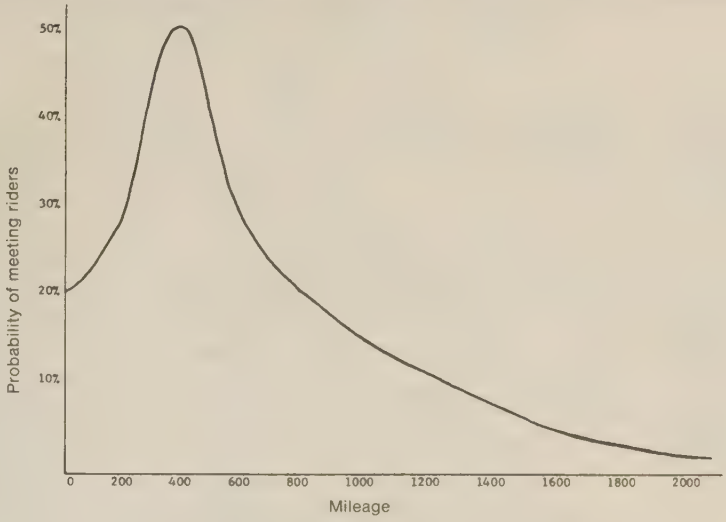
  Table 1 shows a frequency analysis of events mentioned in the diaries of three people that traveled the entire length of the trail. The probabilities of events occurring in the simulation are based on this analysis.

- Miscellaneous

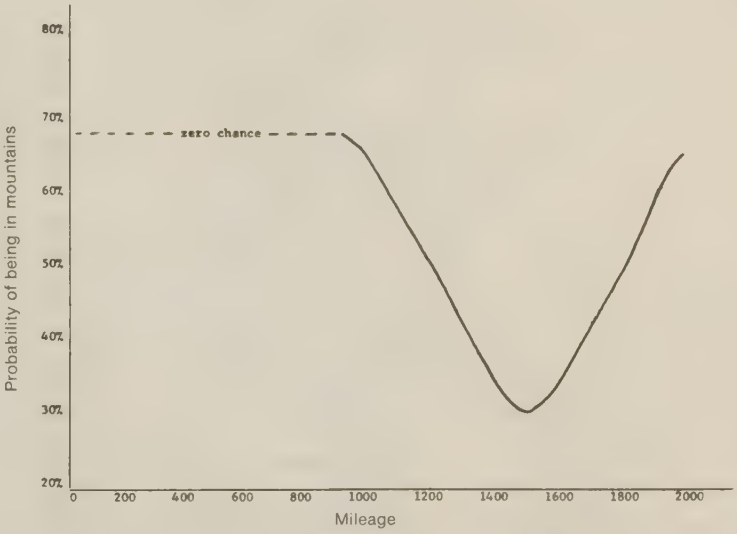
  Dates and days of the week shown are correct for 1847.

  The average trip in the simulation takes about 12 two-week turns. There were six forts on the trail. In the simulation a player gets the option to stop at a fort every other turn.

Probability curves for being attacked by riders and for being in the mountains are representative of the geographic features of the land. (Riders attack more frequently on the plains.)



Occurrence of "Riders Ahead" as a function of mileage



Occurrence of "Rugged Mountains" as a function of mileage

BIBLIOGRAPHY

Ghent, William J., *The Road to Oregon*, Longmans, Green & Co., New York, 1929

Hancock, Samuel, *Narrative of Samuel Hancock*, George H. Harrap & Co Ltd., London, 1927

Meeker, Ezra, *Ox Team Days on the Oregon Trail*, pub. by E. Meeker, New York, 1907.

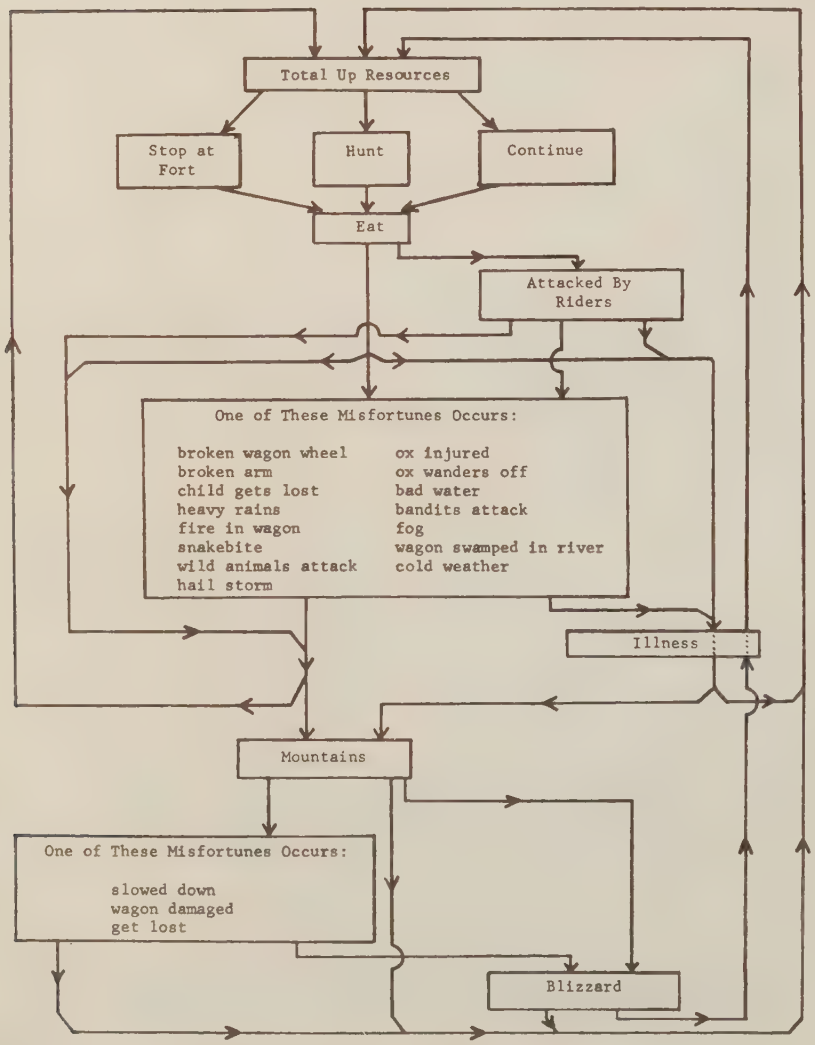
Morgan, Dale L., *Overland in 1846*, Talisman Press, Georgetown, California, 1963

Extensive additional material; sources, and background are contained in the MECC OREGON User Manual by Don Rawitsch Minnesota Educational Computer Consortium, 2520 Broadway Drive, Lauderdale, MN 55113

INFORMATION FROM DIARIES OF PEOPLE TRAVELING THE OREGON TRAIL
(Source: Morgan, David L., *Overland in 1846*, pp. 123-130,150-158,163-188)

Reference	Start Date	End Date	Total Weeks	No. of Occurrences	Frequency of Occurrences			
Diary of William E. Taylor	4/20/1846	9/13/1846	21	32	bad weather	10	illness	4
					rugged trail	4	no water	3
					animals lost	1	injury	0
					wagon breakdown	2	bad river	1
					met friendly Indians	5	thieves	2
Diary of Nicholas Carriger	4/27/1846	9/26/1846	22	30	bad weather	2	illness	10
					rugged trail	6	injury	3
					animals lost	6	thieves	1
					wagon breakdown	2		
Diary of Virgil Pringle	4/15/1846	11/30/1846	33	38	bad weather	7	no water	5
					illness	5	injury	2
					rugged trail	7	bad river	3
					animals lost	3	thieves	2
					wagon breakdown	4		
		TOTALS: ave. 25		ave. 33	bad weather	20%	illness	20%
					rugged trail	16%	no water	8%
					animals lost	10%	injury	5%
					wagon breakdown	8%	bad river	4%
					met friendly Indians	5%	thieves	4%

Table 1



Background on MECC

The Minnesota Educational Computing Consortium (MECC) was created in 1972 out of concern by the governor and legislature that educational computing needed a central source of coordination for planning, and a mechanism to insure that all educational institutions in the state would have equal opportunity of access to computing services for both instructional and administrative programs. The Consortium's membership includes the University of Minnesota (5 campuses), The Minnesota State University System (7 campuses), the Minnesota Community College System (18 campuses), the Minnesota Department of Education (representing the state's 436 independent school districts), and the Minnesota Department of Administration. Minnesota is the only state in the country having a central organization for coordinating educational computing activities across all levels of education.

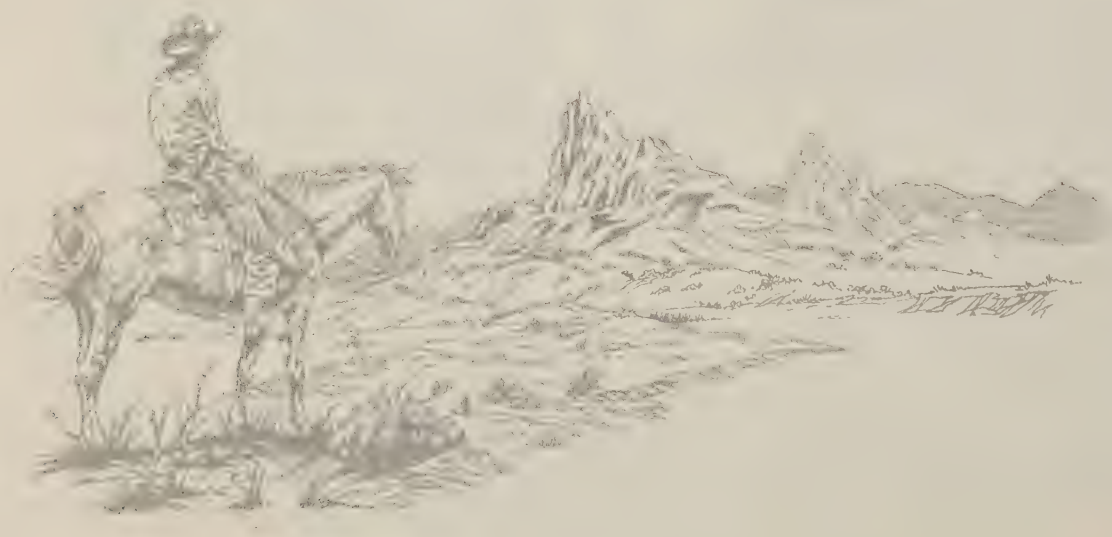
The MECC Instructional Services Division offers a variety of services to consortium members. A technical staff operates the largest of Minnesota's computers dedicated to instructional computing, a Control Data CYBER 73 time-sharing system. The MECC Timeshare System is currently configured for 375 user ports and serves about 1100 interactive terminals located in schools and colleges across the state. A large multiplexing communications network provides the route by which MECC users access the Timeshare System, whether they are a few miles from the Minneapolis-St. Paul computer center or hundreds of miles away near the Canadian border. The MECC User Services staff of instructional coordinators helps user learn to make better use of the computer by visiting school and college sites, conducting workshops, providing over-the-phone consulting service, publishing news letters, and producing written documentation for programs in the MECC Timeshare System's central library.

OREGON: Detailed Model

Sample Run

```
THIS PROGRAM SIMULATES A TRIP OVER THE OREGON TRAIL FROM
INDEPENDENCE, MISSOURI TO OREGON CITY, OREGON IN 1847.
YOUR FAMILY OF FIVE WILL COVER THE 2040 MILE OREGON TRAIL
IN 5-6 MONTHS --- IF YOU MAKE IT ALIVE.

YOU HAD SAVED $900 TO SPEND FOR THE TRIP, AND YOU'VE JUST
   PAID $200 FOR A WAGON.
YOU WILL NEED TO SPEND THE REST OF YOUR MONEY ON THE
   FOLLOWING ITEMS

     OXEN - YOU CAN SPEND $200-$300 ON YOUR TEAM
            THE MORE YOU SPEND, THE FASTER YOU'LL GO
               BECAUSE YOU'LL HAVE BETTER ANIMALS

     FOOD - THE MORE YOU HAVE, THE LESS CHANCE THERE
               IS OF GETTING SICK

     AMMUNITION - $1 BUYS A BELT OF 50 BULLETS
            YOU WILL NEED BULLETS FOR ATTACKS BY ANIMALS
               AND BANDITS, AND FOR HUNTING FOOD

     CLOTHING - THIS IS ESPECIALLY IMPORTANT FOR THE COLD
               WEATHER YOU WILL ENCOUNTER WHEN CROSSING
               THE MOUNTAINS

     MISCELLANEOUS SUPPLIES - THIS INCLUDES MEDICINE AND
               OTHER THINGS YOU WILL NEED FOR SICKNESS
               AND EMERGENCY REPAIRS


YOU CAN SPEND ALL YOUR MONEY BEFORE YOU START YOUR TRIP -
OR YOU CAN SAVE SOME OF YOUR CASH TO SPEND AT FORTS ALONG
THE WAY WHEN YOU RUN LOW.  HOWEVER, ITEMS COST MORE AT
THE FORTS.  YOU CAN ALSO GO HUNTING ALONG THE WAY TO GET
MORE FOOD.
WHENEVER YOU HAVE TO USE YOUR TRUSTY RIFLE ALONG THE WAY,
YOU WILL BE TOLD TO TYPE IN A WORD (ONE THAT SOUNDS LIKE A
GUN SHOT).  THE FASTER YOU TYPE IN THAT WORD AND HIT THE
"RETURN" KEY, THE BETTER LUCK YOU'LL HAVE WITH YOUR GUN.

AT EACH TURN, ALL ITEMS ARE SHOWN IN DOLLAR AMOUNTS
EXCEPT BULLETS
WHEN ASKED TO ENTER MONEY AMOUNTS, DON'T USE A "$".

GOOD LUCK!!!


HOW GOOD A SHOT ARE YOU WITH YOUR RIFLE?
  (1) ACE MARKSMAN,  (2) GOOD SHOT,  (3) FAIR TO MIDDLIN'
        (4) NEED MORE PRACTICE,  (5) SHAKY KNEES
ENTER ONE OF THE ABOVE -- THE BETTER YOU CLAIM YOU ARE, THE
FASTER YOU'LL HAVE TO BE WITH YOUR GUN TO BE SUCCESSFUL.
? 4


HOW MUCH DO YOU WANT TO SPEND ON YOUR OXEN TEAM ? 250
HOW MUCH DO YOU WANT TO SPEND ON FOOD ? 150
HOW MUCH DO YOU WANT TO SPEND ON AMMUNITION ? 50
HOW MUCH DO YOU WANT TO SPEND ON CLOTHING ? 150
HOW MUCH DO YOU WANT TO SPEND ON MISCELLANEOUS SUPPLIES ? 50
AFTER ALL YOUR PURCHASES, YOU NOW HAVE  50  DOLLARS LEFT


MONDAY MARCH 29 1847

TOTAL MILEAGE IS 0
FOOD           BULLETS        CLOTHING       MISC. SUPP.    CASH
 150            2500           150            50             50
DO YOU WANT TO (1) HUNT, OR (2) CONTINUE
? 2
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 1
THERE WAS A FIRE IN YOUR WAGON--FOOD AND SUPPLIES DAMAGED

MONDAY APRIL 12 1847

TOTAL MILEAGE IS 200
FOOD           BULLETS        CLOTHING       MISC. SUPP.    CASH
 97             2100           150            40             50
DO YOU WANT TO (1) STOP AT THE NEXT FORT, (2) HUNT, OR (3) CONTINUE
? 3
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 1
YOU KILLED A POISONOUS SNAKE AFTER IT BIT YOU

MONDAY APRIL 26 1847

TOTAL MILEAGE IS 409
FOOD           BULLETS        CLOTHING       MISC. SUPP.    CASH
 84             2090           150            35             50
DO YOU WANT TO (1) HUNT, OR (2) CONTINUE
? 2
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2
WAGON GETS SWAMPED FORDING RIVER--LOSE FOOD AND CLOTHES

MONDAY MAY 10 1847

TOTAL MILEAGE IS 580
FOOD           BULLETS        CLOTHING       MISC. SUPP.    CASH
 36             2090           130            35             50
DO YOU WANT TO (1) STOP AT THE NEXT FORT, (2) HUNT, OR (3) CONTINUE
? 2
TYPE WHAM
? WHAM

RIGHT BETWEEN THE EYES---YOU GOT A BIG ONE!!!!
FULL BELLIES TONIGHT!
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 3
WAGON GETS SWAMPED FORDING RIVER--LOSE FOOD AND CLOTHES


MONDAY MAY 24 1847

TOTAL MILEAGE IS 719
FOOD           BULLETS        CLOTHING       MISC. SUPP.    CASH
 36             2078           110            35             50
DO YOU WANT TO (1) HUNT, OR (2) CONTINUE
? 1
TYPE POW
? POW

RIGHT BETWEEN THE EYES---YOU GOT A BIG ONE!!!!
FULL BELLIES TONIGHT!
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2
RIDERS AHEAD.  THEY LOOK HOSTILE
TACTICS
(1) RUN  (2) ATTACK  (3) CONTINUE  (4) CIRCLE WAGONS
? 2
TYPE BLAM
? BLAM

NICE SHOOTING---YOU DROVE THEM OFF
RIDERS WERE HOSTILE--CHECK FOR LOSSES
HELPFUL INDIANS SHOW YOU WHERE TO FIND MORE FOOD

MONDAY JUNE 7 1847

TOTAL MILEAGE IS 883
FOOD           BULLETS        CLOTHING       MISC. SUPP.    CASH
 86             1984           110            35             50
DO YOU WANT TO (1) STOP AT THE NEXT FORT, (2) HUNT, OR (3) CONTINUE
? 3
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 1
WAGON GETS SWAMPED FORDING RIVER--LOSE FOOD AND CLOTHES
RUGGED MOUNTAINS
THE GOING GETS SLOW
BLIZZARD IN MOUNTAIN PASS--TIME AND SUPPLIES LOST

MONDAY JUNE 21 1847

TOTAL MILEAGE IS 950
FOOD           BULLETS        CLOTHING       MISC. SUPP.    CASH
 18             1684           90             25             50
DO YOU WANT TO (1) HUNT, OR (2) CONTINUE
? 1
TYPE BLAM
? BLAM

RIGHT BETWEEN THE EYES---YOU GOT A BIG ONE!!!!
FULL BELLIES TONIGHT!
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2
RIDERS AHEAD.  THEY LOOK HOSTILE
TACTICS
(1) RUN  (2) ATTACK  (3) CONTINUE  (4) CIRCLE WAGONS
```

```
? 2
TYPE BLAM
? BLAM

NICE SHOOTING---YOU DROVE THEM OFF
RIDERS WERE HOSTILE--CHECK FOR LOSSES
HAIL STORM---SUPPLIES DAMAGED
RUGGED MOUNTAINS
THE GOING GETS SLOW

MONDAY JULY 5 1847

TOTAL MILEAGE IS 1004
FOOD          BULLETS        CLOTHING       MISC. SUPP.    CASH
 53            1392           90             19             50
DO YOU WANT TO (1) STOP AT THE NEXT FORT, (2) HUNT, OR (3) CONTINUE
? 3
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2
WILD ANIMALS ATTACK!
TYPE POW
? POW

NICE SHOOTIN' PARDNER---THEY DIDN'T GET MUCH

MONDAY JULY 19 1847

TOTAL MILEAGE IS 1217
FOOD          BULLETS        CLOTHING       MISC. SUPP.    CASH
 35            1392           90             19             50
DO YOU WANT TO (1) HUNT, OR (2) CONTINUE
? 2
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2


MONDAY AUGUST 2 1847

TOTAL MILEAGE IS 1429
FOOD          BULLETS        CLOTHING       MISC. SUPP.    CASH
 17            1392           90             19             50
DO YOU WANT TO (1) STOP AT THE NEXT FORT, (2) HUNT, OR (3) CONTINUE
? 2
TYPE BANG
? BANG

RIGHT BETWEEN THE EYES---YOU GOT A BIG ONE!!!!
FULL BELLIES TONIGHT!
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2

MONDAY AUGUST 16 1847

TOTAL MILEAGE IS 1594
FOOD          BULLETS        CLOTHING       MISC. SUPP.    CASH
 55            1380           90             19             50
DO YOU WANT TO (1) HUNT, OR (2) CONTINUE
? 2
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2

RIDERS AHEAD.  THEY LOOK HOSTILE
TACTICS
(1) RUN  (2) ATTACK  (3) CONTINUE  (4) CIRCLE WAGONS
? 2
TYPE WHAM
? WHAM

NICE SHOOTING---YOU DROVE THEM OFF
RIDERS WERE HOSTILE--CHECK FOR LOSSES
WILD ANIMALS ATTACK!
TYPE BLAM
? BLAM

NICE SHOOTIN' PARDNER---THEY DIDN'T GET MUCH
RUGGED MOUNTAINS
THE GOING GETS SLOW
BLIZZARD IN MOUNTAIN PASS--TIME AND SUPPLIES LOST

MONDAY AUGUST 31 1847

YOU'D BETTER DO SOME HUNTING OR BUY FOOD AND SOON!!!!
TOTAL MILEAGE IS 1685
FOOD          BULLETS        CLOTHING       MISC. SUPP.    CASH
 11            999            89             9              50
DO YOU WANT TO (1) STOP AT THE NEXT FORT, (2) HUNT, OR (3) CONTINUE
? 2
TYPE POW
? POW

RIGHT BETWEEN THE EYES---YOU GOT A BIG ONE!!!!
FULL BELLIES TONIGHT!
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 2

MONDAY SEPTEMBER 13 1847

TOTAL MILEAGE IS 1847
FOOD          BULLETS        CLOTHING       MISC. SUPP.    CASH
 48            986            89             9              50
DO YOU WANT TO (1) HUNT, OR (2) CONTINUE
? 2
DO YOU WANT TO EAT (1) POORLY  (2) MODERATELY
OR (3) WELL ? 1
COLD WEATHER---BRRRRRRR!---YOU HAVE ENOUGH CLOTHING TO KEEP YOU WARM

YOU FINALLY ARRIVED AT OREGON CITY
AFTER 2040 LONG MILES---HOORAY!!!!!
A REAL PIONEER!

SATURDAY SEPTEMBER  25  1847

FOOD          BULLETS        CLOTHING       MISC. SUPP.    CASH
 35            986            89             9              50

          PRESIDENT JAMES K. POLK SENDS YOU HIS
               HEARTIEST CONGRATULATIONS

          AND WISHES YOU A PROSPEROUS LIFE AHEAD

                    AT YOUR NEW HOME
```



Program Listing

```
 10 REM PROGRAM NAME - OREGON          VERSION:01/01/78
 20 REM ORIGINAL PROGRAMMING BY BILL HEINEMANN - 1971
 30 REM SUPPORT RESEARCH AND MATERIALS BY DON RAWITSCH,
 40 REM      MINNESOTA EDUCATIONAL COMPUTING CONSORTIUM STAFF
 50 REM CDC CYBER 70/73-26      BASIC 3.1
 60 REM DOCUMENTATION BOOKLET 'OREGON' AVAILABLE FROM
 61 REM     MECC SUPPORT SERVICES
 62 REM     2520 BROADWAY DRIVE
 63 REM     ST. PAUL, MN  55113
 80 REM
150 REM  *FOR THE MEANING OF THE VARIABLES USED, LIST LINES 6470-6790*
155 REM
160 PRINT "DO YOU NEED INSTRUCTIONS  (YES/NO)";
170 DIM C$(5)
180 REM RANDOMIZE REMOVED
190 INPUT C$
200 IF C$="NO" THEN 690
210 PRINT
220 PRINT
230 REM ***INSTRUCTIONS***
240 PRINT "THIS PROGRAM SIMULATES A TRIP OVER THE OREGON TRAIL FROM"
250 PRINT "INDEPENDENCE, MISSOURI TO OREGON CITY, OREGON IN 1847."
260 PRINT "YOUR FAMILY OF FIVE WILL COVER THE 2040 MILE OREGON TRAIL"
270 PRINT "IN 5-6 MONTHS --- IF YOU MAKE IT ALIVE."
280 PRINT
290 PRINT "YOU HAD SAVED $900 TO SPEND FOR THE TRIP, AND YOU'VE JUST"
300 PRINT "   PAID $200 FOR A WAGON."
310 PRINT "YOU WILL NEED TO SPEND THE REST OF YOUR MONEY ON THE"
320 PRINT "   FOLLOWING ITEMS:"
330 PRINT
340 PRINT "     OXEN - YOU CAN SPEND $200-$300 ON YOUR TEAM"
350 PRINT "            THE MORE YOU SPEND, THE FASTER YOU'LL GO"
360 PRINT "               BECAUSE YOU'LL HAVE BETTER ANIMALS"
370 PRINT
380 PRINT "     FOOD - THE MORE YOU HAVE, THE LESS CHANCE THERE"
390 PRINT "               IS OF GETTING SICK"
400 PRINT
410 PRINT "     AMMUNITION - $1 BUYS A BELT OF 50 BULLETS"
420 PRINT "            YOU WILL NEED BULLETS FOR ATTACKS BY ANIMALS"
430 PRINT "               AND BANDITS, AND FOR HUNTING FOOD"
440 PRINT
450 PRINT "     CLOTHING - THIS IS ESPECIALLY IMPORTANT FOR THE COLD"
460 PRINT "               WEATHER YOU WILL ENCOUNTER WHEN CROSSING"
470 PRINT "               THE MOUNTAINS"
480 PRINT
490 PRINT "     MISCELLANEOUS SUPPLIES - THIS INCLUDES MEDICINE AND"
500 PRINT "               OTHER THINGS YOU WILL NEED FOR SICKNESS"
510 PRINT "               AND EMERGENCY REPAIRS"
520 PRINT
530 PRINT
540 PRINT "YOU CAN SPEND ALL YOUR MONEY BEFORE YOU START YOUR TRIP -"
550 PRINT "OR YOU CAN SAVE SOME OF YOUR CASH TO SPEND AT FORTS ALONG"
560 PRINT "THE WAY WHEN YOU RUN LOW.  HOWEVER, ITEMS COST MORE AT"
570 PRINT "THE FORTS.  YOU CAN ALSO GO HUNTING ALONG THE WAY TO GET"
580 PRINT "MORE FOOD."
590 PRINT "WHENEVER YOU HAVE TO USE YOUR TRUSTY RIFLE ALONG THE WAY,"
600 PRINT "YOU WILL BE TOLD TO TYPE IN A WORD (ONE THAT SOUNDS LIKE A"
610 PRINT "GUN SHOT).  THE FASTER YOU TYPE IN THAT WORD AND HIT THE"
620 PRINT ""RETURN"" KEY, THE BETTER LUCK YOU'LL HAVE WITH YOUR GUN."
630 PRINT
640 PRINT "AT EACH TURN, ALL ITEMS ARE SHOWN IN DOLLAR AMOUNTS"
650 PRINT "EXCEPT BULLETS"
660 PRINT "WHEN ASKED TO ENTER MONEY AMOUNTS, DON'T USE A ""$""."
670 PRINT
680 PRINT "GOOD LUCK!!!"
690 PRINT
700 PRINT
710 PRINT "HOW GOOD A SHOT ARE YOU WITH YOUR RIFLE?"
720 PRINT "  (1) ACE MARKSMAN,  (2) GOOD SHOT,  (3) FAIR TO MIDDLIN'"
730 PRINT "        (4) NEED MORE PRACTICE,  (5) SHAKY KNEES"
740 PRINT "ENTER ONE OF THE ABOVE -- THE BETTER YOU CLAIM YOU ARE, THE"
750 PRINT "FASTER YOU'LL HAVE TO BE WITH YOUR GUN TO BE SUCCESSFUL."
760 INPUT D9
770 IF D9>5 THEN 790
780 GOTO 810
790 D9=0
800 REM ***INITIAL PURCHASES***
810 X1=-1
820 K8=S4=F1=F2=M=M9=D3=0
830 PRINT
840 PRINT
850 PRINT "HOW MUCH DO YOU WANT TO SPEND ON YOUR OXEN TEAM";
860 INPUT A
870 IF A >= 200 THEN 900
880 PRINT "NOT ENOUGH"
890 GOTO 850
900 IF A <= 300 THEN 930
910 PRINT "TOO MUCH"
920 GOTO 850
930 PRINT "HOW MUCH DO YOU WANT TO SPEND ON FOOD";
940 INPUT F
950 IF F >= 0 THEN 980
960 PRINT "IMPOSSIBLE"
970 GOTO 930
980 PRINT "HOW MUCH DO YOU WANT TO SPEND ON AMMUNITION";
990 INPUT B
1000 IF B >= 0 THEN 1030
1010 PRINT "IMPOSSIBLE"
1020 GOTO 980
1030 PRINT "HOW MUCH DO YOU WANT TO SPEND ON CLOTHING";
1040 INPUT C
1050 IF C >= 0 THEN 1080
1060 PRINT "IMPOSSIBLE"
1070 GOTO 1030
1080 PRINT "HOW MUCH DO YOU WANT TO SPEND ON MISCELLANEOUS SUPPLIES";
1090 INPUT M1
1100 IF M1 >= 0 THEN 1130
1110 PRINT "IMPOSSIBLE"
1120 GOTO 1080
1130 T=700-A-F-B-C-M1
1140 IF T >= 0 THEN 1170
1150 PRINT "YOU OVERSPENT--YOU ONLY HAD $700 TO SPEND.  BUY AGAIN"
1160 GOTO 830
1170 B=50*B
1180 PRINT "AFTER ALL YOUR PURCHASES, YOU NOW HAVE";T;"DOLLARS LEFT"
1190 PRINT
1200 PRINT "MONDAY MARCH 29 1847"
1210 PRINT
1220 GOTO 1750
1230 IF M >= 2040 THEN 5430
1240 REM ***SETTING DATE***
1250 D3=D3+1
1260 PRINT
1270 PRINT "MONDAY ";
1280 IF D3>10 THEN 1300
1290 ON D3 GOTO 1310,1330,1350,1370,1390,1410,1430,1450,1470,1490
1300 ON D3-10 GOTO 1510,1530,1550,1570,1590,1610,1630,1650,1670,1690
1310 PRINT "APRIL 12 ";
1320 GOTO 1720
1330 PRINT "APRIL 26 ";
1340 GOTO 1720
1350 PRINT "MAY 10 ";
1360 GOTO 1720
1370 PRINT "MAY 24 ";
1380 GOTO 1720
1390 PRINT "JUNE 7 ";
1400 GOTO 1720
1410 PRINT "JUNE 21 ";
1420 GOTO 1720
1430 PRINT "JULY 5 ";
1440 GOTO 1720
1450 PRINT "JULY 19 ";
1460 GOTO 1720
1470 PRINT "AUGUST 2 ";
1480 GOTO 1720
1490 PRINT "AUGUST 16 ";
1500 GOTO 1720
1510 PRINT "AUGUST 31 ";
1520 GOTO 1720
1530 PRINT "SEPTEMBER 13 ";
1540 GOTO 1720
1550 PRINT "SEPTEMBER 27 ";
1560 GOTO 1720
1570 PRINT "OCTOBER 11 ";
1580 GOTO 1720
1590 PRINT "OCTOBER 25 ";
1600 GOTO 1720
1610 PRINT "NOVEMBER 8 ";
1620 GOTO 1720
1630 PRINT "NOVEMBER 22 ";
1640 GOTO 1720
1650 PRINT "DECEMBER 6 ";
1660 GOTO 1720
1670 PRINT "DECEMBER 20 ";
1680 GOTO 1720
1690 PRINT "YOU HAVE BEEN ON THE TRAIL TOO LONG  ------"
1700 PRINT "YOUR FAMILY DIES IN THE FIRST BLIZZARD OF WINTER"
1710 GOTO 5170
1720 PRINT "1847"
1730 PRINT
1740 REM ***BEGINNING EACH TURN***
1750 IF F >= 0 THEN 1770
1760 F=0
1770 IF B >= 0 THEN 1790
1780 B=0
1790 IF C >= 0 THEN 1810
1800 C=0
1810 IF M1 >= 0 THEN 1830
1820 M1=0
1830 IF F >= 13 THEN 1850
1840 PRINT "YOU'D BETTER DO SOME HUNTING OR BUY FOOD AND SOON!!!!"
1850 F=INT(F)
1860 B=INT(B)
1870 C=INT(C)
1880 M1=INT(M1)
1890 T=INT(T)
1900 M=INT(M)
1910 M2=M
1920 IF S4=1 THEN 1950
1930 IF K8=1 THEN 1950
1940 GOTO 1990
1950 T=T-20
1960 IF T<0 THEN 5080
1970 PRINT "DOCTOR'S BILL IS $20"
1980 LET K8=S4=0
1990 IF M9=1 THEN 2020
2000 PRINT "TOTAL MILEAGE IS";M
2010 GOTO 2040
2020 PRINT "TOTAL MILEAGE IS 950"
2030 M9=0
2040 PRINT "FOOD","BULLETS","CLOTHING","MISC. SUPP.","CASH"
2050 PRINT F,B,C,M1,T
2060 IF X1=-1 THEN 2170
2070 X1=X1*(-1)
2080 PRINT "DO YOU WANT TO (1) STOP AT THE NEXT FORT, (2) HUNT, "
2090 PRINT "OR (3) CONTINUE"
2100 INPUT X
2110 IF X>2 THEN 2150
2120 IF X<1 THEN 2150
2130 LET X=INT(X)
```

```
2140 GØTØ 2270
2150 LET X=3
2160 GØTØ 2270
2170 PRINT "DØ YØU WANT TØ (1) HUNT, ØR (2) CØNTINUE"
2180 INPUT X
2190 IF X=1 THEN 2210
2200 LET X=2
2210 LET X=X+1
2220 IF X=3 THEN 2260
2230 IF B>39 THEN 2260
2240 PRINT "TØUGH---YØU NEED MØRE BULLETS TØ GØ HUNTING"
2250 GØTØ 2170
2260 X1=X1*(-1)
2270 ØN X GØTØ 2290,2540,2720
2280 REM ***STØPPING AT FØRT***
2290 PRINT "ENTER WHAT YØU WISH TØ SPEND ØN THE FØLLØWING"
2300 PRINT "FØØD";
2310 GØSUB 2330
2320 GØTØ 2410
2330 INPUT P
2340 IF P<0 THEN 2400
2350 T=T-P
2360 IF T >= 0 THEN 2400
2370 PRINT "YØU DØN'T HAVE THAT MUCH--KEEP YØUR SPENDING DØWN"
2375 PRINT "YØU MISS YØUR CHANCE TØ SPEND ØN THAT ITEM"
2380 T=T+P
2390 P=0
2400 RETURN
2410 F=F+2/3*P
2420 PRINT "AMMUNITIØN";
2430 GØSUB 2330
2440 LET B=INT(B+2/3*P*50)
2450 PRINT "CLØTHING";
2460 GØSUB 2330
2470 C=C+2/3*P
2480 PRINT "MISCELLANEØUS SUPPLIES";
2490 GØSUB 2330
2500 M1=M1+2/3*P
2510 M=M-45
2520 GØTØ 2720
2530 REM ***HUNTING***
2540 IF B>39 THEN 2570
2550 PRINT "TØUGH---YØU NEED MØRE BULLETS TØ GØ HUNTING"
2560 GØTØ 2080
2570 M=M-45
2580 GØSUB 6140
2590 IF B1 <= 1 THEN 2660
2600 IF 100*RND(-1)<13*B1 THEN 2710
2610 F=F+48-2*B1
2620 PRINT "NICE SHØT--RIGHT ØN TARGET--GØØD EATIN' TØNIGHT!!"
2630 B=B-10-3*B1
2640 GØTØ 2720
2650 REM **BELLS IN LINE 2660**
2660 PRINT "RIGHT BETWEEN THE EYES---YØU GØT A BIG ØNE!!!!"
2670 PRINT "FULL BELLIES TØNIGHT!"
2680 F=F+52+RND(-1)*6
2690 B=B-10-RND(-1)*4
2700 GØTØ 2720
2710 PRINT "YØU MISSED---AND YØUR DINNER GØT AWAY....."
2720 IF F >= 13 THEN 2750
2730 GØTØ 5060
2740 REM ***EATING***
2750 PRINT "DØ YØU WANT TØ EAT (1) PØØRLY  (2) MØDERATELY"
2760 PRINT "ØR (3) WELL";
2770 INPUT E
2780 IF E>3 THEN 2750
2790 IF E<1 THEN 2750
2800 LET E=INT(E)
2810 LET F=F-8-5*E
2820 IF F >= 0 THEN 2860
2830 F=F+8+5*E
2840 PRINT "YØU CAN'T EAT THAT WELL"
2850 GØTØ 2750
2860 LET M=M+200+(A-220)/5+10*RND(-1)
2870 L1=C1=0
2880 REM ***RIDERS ATTACK***
2890 IF RND(-1)*10>((M/100-4)**2+72)/((M/100-4)**2+12)-1 THEN 3550
2900 PRINT "RIDERS AHEAD.  THEY ";
2910 S5=0
2920 IF RND(-1)<.8 THEN 2950
2930 PRINT "DØN'T ";
2940 S5=1
2950 PRINT "LØØK HØSTILE"
2960 PRINT "TACTICS"
2970 PRINT "(1) RUN  (2) ATTACK  (3) CØNTINUE  (4) CIRCLE WAGØNS"
2980 IF RND(-1)>.2 THEN 3000
2990 S5=1-S5
3000 INPUT T1
3010 IF T1<1 THEN 2970
3020 IF T1>4 THEN 2970
3030 T1=INT(T1)
3040 IF S5=1 THEN 3330
3050 IF T1>1 THEN 3110
3060 M=M+20
3070 M1=M1-15
3080 B=B-150
3090 A=A-40
3100 GØTØ 3470
3110 IF T1>2 THEN 3240
3120 GØSUB 6140
3130 B=B-B1*40-80
3140 IF B1>1 THEN 3170
3150 PRINT "NICE SHØØTING---YØU DRØVE THEM ØFF"
3160 GØTØ 3470
3170 IF B1 <= 4 THEN 3220
3180 PRINT "LØUSY SHØT---YØU GØT KNIFED"
3190 K8=1
3200 PRINT "YØU HAVE TØ SEE ØL' DØC BLANCHARD"
3210 GØTØ 3470
3220 PRINT "KINDA SLØW WITH YØUR CØLT .45"
3230 GØTØ 3470
3240 IF T1>3 THEN 3290
3250 IF RND(-1)>.8 THEN 3450
3260 LET B=B-150
3270 M1=M1-15
3280 GØTØ 3470
3290 GØSUB 6140
3300 B=B-B1*30-80
3310 M=M-25
3320 GØTØ 3140
3330 IF T1>1 THEN 3370
3340 M=M+15
3350 A=A-10
3360 GØTØ 3470
3370 IF T1>2 THEN 3410
3380 M=M-5
3390 B=B-100
3400 GØTØ 3470
3410 IF T1>3 THEN 3430
3420 GØTØ 3470
3430 M=M-20
3440 GØTØ 3470
3450 PRINT "THEY DID NØT ATTACK"
3460 GØTØ 3550
3470 IF S5=0 THEN 3500
3480 PRINT "RIDERS WERE FRIENDLY, BUT CHECK FØR PØSSIBLE LØSSES"
3490 GØTØ 3550
3500 PRINT "RIDERS WERE HØSTILE--CHECK FØR LØSSES"
3510 IF B >= 0 THEN 3550
3520 PRINT "YØU RAN ØUT ØF BULLETS AND GØT MASSACRED BY THE RIDERS"
3530 GØTØ 5170
3540 REM ***SELECTIØN ØF EVENTS***
3550 LET D1=0
3560 RESTØRE
3570 R1=100*RND(-1)
3580 LET D1=D1+1
3590 IF D1=16 THEN 4670
3600 READ D
3610 IF R1>D THEN 3580
3620 DATA 6,11,13,15,17,22,32,35,37,42,44,54,64,69,95
3630 IF D1>10 THEN 3650
3640 ØN D1 GØTØ 3660,3700,3740,3790,3820,3850,3880,3960,4130,4190
3650 ØN D1-10 GØTØ 4220,4290,4340,4560,4610,4670
3660 PRINT "WAGØN BREAKS DØWN--LØSE TIME AND SUPPLIES FIXING IT"
3670 LET M=M-15-5*RND(-1)
3680 LET M1=M1-8
3690 GØTØ 4710
3700 PRINT "ØX INJURES LEG---SLØWS YØU DØWN REST ØF TRIP"
3710 LET M=M-25
3720 LET A=A-20
3730 GØTØ 4710
3740 PRINT "BAD LUCK---YØUR DAUGHTER BRØKE HER ARM"
3750 PRINT "YØU HAD TØ STØP AND USE SUPPLIES TØ MAKE A SLING"
3760 M=M-5-4*RND(-1)
3770 M1=M1-2-3*RND(-1)
3780 GØTØ 4710
3790 PRINT "ØX WANDERS ØFF---SPEND TIME LØØKING FØR IT"
3800 M=M-17
3810 GØTØ 4710
3820 PRINT "YØUR SØN GETS LØST---SPEND HALF THE DAY LØØKING FØR HIM"
3830 M=M-10
3840 GØTØ 4710
3850 PRINT "UNSAFE WATER--LØSE TIME LØØKING FØR CLEAN SPRING"
3860 LET M=M-10*RND(-1)-2
3870 GØTØ 4710
3880 IF M>950 THEN 4490
3890 PRINT "HEAVY RAINS---TIME AND SUPPLIES LØST"
3910 F=F-10
3920 B=B-500
3930 M1=M1-15
3940 M=M-10*RND(-1)-5
3950 GØTØ 4710
3960 PRINT "BANDITS ATTACK"
3970 GØSUB 6140
3980 B=B-20*B1
3990 IF B >= 0 THEN 4030
4000 PRINT "YØU RAN ØUT ØF BULLETS---THEY GET LØTS ØF CASH"
4010 T=T/3
4020 GØTØ 4040
4030 IF B1 <= 1 THEN 4100
4040 PRINT "YØU GØT SHØT IN THE LEG AND THEY TØØK ØNE ØF YØUR ØXEN"
4050 K8=1
4060 PRINT "BETTER HAVE A DØC LØØK AT YØUR WØUND"
4070 M1=M1-5
4080 A=A-20
4090 GØTØ 4710
4100 PRINT "QUICKEST DRAW ØUTSIDE ØF DØDGE CITY!!!"
4110 PRINT "YØU GØT 'EM!"
4120 GØTØ 4710
4130 PRINT "THERE WAS A FIRE IN YØUR WAGØN--FØØD AND SUPPLIES DAMAGED"
4140 F=F-40
4150 B=B-400
4160 LET M1=M1-RND(-1)*8-3
4170 M=M-15
4180 GØTØ 4710
4190 PRINT "LØSE YØUR WAY IN HEAVY FØG---TIME IS LØST"
4200 M=M-10-5*RND(-1)
4210 GØTØ 4710
4220 PRINT "YØU KILLED A PØISØNØUS SNAKE AFTER IT BIT YØU"
4230 B=B-10
4240 M1=M1-5
4250 IF M1 >= 0 THEN 4280
4260 PRINT "YØU DIE ØF SNAKEBITE SINCE YØU HAVE NØ MEDICINE"
4270 GØTØ 5170
4280 GØTØ 4710
4290 PRINT "WAGØN GETS SWAMPED FØRDING RIVER--LØSE FØØD AND CLØTHES"
4300 F=F-30
4310 C=C-20
4320 M=M-20-20*RND(-1)
4330 GØTØ 4710
4340 PRINT "WILD ANIMALS ATTACK!"
4350 GØSUB 6140
4360 IF B>39 THEN 4410
4370 PRINT "YØU WERE TØØ LØW ØN BULLETS--"
4380 PRINT "THE WØLVES ØVERPØWERED YØU"
4390 K8=1
4400 GØTØ 5120
4410 IF B1>2 THEN 4440
4420 PRINT "NICE SHØØTIN' PARDNER---THEY DIDN'T GET MUCH"
4430 GØTØ 4450
4440 PRINT "SLØW ØN THE DRAW---THEY GØT AT YØUR FØØD AND CLØTHES"
4450 B=B-20*B1
4460 C=C-B1*4
4470 F=F-B1*8
4480 GØTØ 4710
4490 PRINT "CØLD WEATHER---BRRRRRRR!---YØU ";
4500 IF C>22+4*RND(-1) THEN 4530
4510 PRINT "DØN'T ";
```

```
4520 C1=1
4530 PRINT "HAVE ENOUGH CLOTHING TO KEEP YOU WARM"
4540 IF C1=0 THEN 4710
4550 GOTO 6300
4560 PRINT "HAIL STORM---SUPPLIES DAMAGED"
4570 M=M-5-RND(-1)*10
4580 B=B-200
4590 M1=M1-4-RND(-1)*3
4600 GOTO 4710
4610 IF E=1 THEN 6300
4620 IF E=3 THEN 4650
4630 IF RND(-1)>.25 THEN 6300
4640 GOTO 4710
4650 IF RND(-1)<.5 THEN 6300
4660 GOTO 4710
4670 PRINT "HELPFUL INDIANS SHOW YOU WHERE TO FIND MORE FOOD"
4680 F=F+14
4690 GOTO 4710
4700 REM ***MOUNTAINS***
4710 IF M <= 950 THEN 1230
4720 IF RND(-1)*10>9-((M/100-15)**2+72)/((M/100-15)**2+12) THEN 4860
4730 PRINT "RUGGED MOUNTAINS"
4740 IF RND(-1)>.1 THEN 4780
4750 PRINT "YOU GOT LOST---LOSE VALUABLE TIME TRYING TO FIND TRAIL!"
4760 M=M-60
4770 GOTO 4860
4780 IF RND(-1)>.11 THEN 4840
4790 PRINT "WAGON DAMAGED!---LOSE TIME AND SUPPLIES"
4800 M1=M1-5
4810 B=B-200
4820 M=M-20-30*RND(-1)
4830 GOTO 4860
4840 PRINT "THE GOING GETS SLOW"
4850 M=M-45-RND(-1)/.02
4860 IF F1=1 THEN 4900
4870 F1=1
4880 IF RND(-1)<.8 THEN 4970
4890 PRINT "YOU MADE IT SAFELY THROUGH SOUTH PASS--NO SNOW"
4900 IF M<1700 THEN 4940
4910 IF F2=1 THEN 4940
4920 F2=1
4930 IF RND(-1)<.7 THEN 4970
4940 IF M>950 THEN 1230
4950 M9=1
4960 GOTO 1230
4970 PRINT "BLIZZARD IN MOUNTAIN PASS--TIME AND SUPPLIES LOST"
4980 L1=1
4990 F=F-25
5000 M1=M1-10
5010 B=B-300
5020 M=M-30-40*RND(-1)
5030 IF C<18+2*RND(-1) THEN 6300
5040 GOTO 4940
5050 REM ***DYING***
5060 PRINT "YOU RAN OUT OF FOOD AND STARVED TO DEATH"
5070 GOTO 5170
5080 LET T=0
5090 PRINT "YOU CAN'T AFFORD A DOCTOR"
5100 GOTO 5120
5110 PRINT "YOU RAN OUT OF MEDICAL SUPPLIES"
5120 PRINT "YOU DIED OF ";
5130 IF K8=1 THEN 5160
5140 PRINT "PNEUMONIA"
5150 GOTO 5170
5160 PRINT "INJURIES"
5170 PRINT
5180 PRINT "DUE TO YOUR UNFORTUNATE SITUATION, THERE ARE A FEW"
5190 PRINT "FORMALITIES WE MUST GO THROUGH"
5200 PRINT
5210 PRINT "WOULD YOU LIKE A MINISTER?"
5220 INPUT C$
5230 PRINT "WOULD YOU LIKE A FANCY FUNERAL?"
5240 INPUT C$
5250 PRINT "WOULD YOU LIKE US TO INFORM YOUR NEXT OF KIN?"
5260 INPUT C$
5270 IF C$="YES" THEN 5310
5280 PRINT "BUT YOUR AUNT SADIE IN ST. LOUIS IS REALLY WORRIED ABOUT YOU"
5290 PRINT
5300 GOTO 5330
5310 PRINT "THAT WILL BE $4.50 FOR THE TELEGRAPH CHARGE."
5320 PRINT
5330 PRINT "WE THANK YOU FOR THIS INFORMATION AND WE ARE SORRY YOU"
5340 PRINT "DIDN'T MAKE IT TO THE GREAT TERRITORY OF OREGON"
5350 PRINT "BETTER LUCK NEXT TIME"
5360 PRINT
5370 PRINT
5380 PRINT TAB(30);"SINCERELY"
5390 PRINT
5400 PRINT TAB(17);"THE OREGON CITY CHAMBER OF COMMERCE"
5410 STOP
5420 REM ***FINAL TURN***
5430 F9=(2040-M2)/(M-M2)
5440 F=F+(1-F9)*(8+5*E)
5450 PRINT
5460 REM **BELLS IN LINES 5470,5480**
5470 PRINT "YOU FINALLY ARRIVED AT OREGON CITY"
5480 PRINT "AFTER 2040 LONG MILES---HOORAY!!!!!"
5490 PRINT "A REAL PIONEER!"
5500 PRINT
5510 F9=INT(F9*14)
5520 D3=D3+14*F9
5530 F9=F9+1
5540 IF F9<8 THEN 5560
5550 F9=F9-7
5560 ON F9 GOTO 5570,5590,5610,5630,5650,5670,5690
5570 PRINT "MONDAY ";
5580 GOTO 5700
5590 PRINT "TUESDAY ";
5600 GOTO 5700
5610 PRINT "WEDNESDAY ";
5620 GOTO 5700
5630 PRINT "THURSDAY ";
5640 GOTO 5700
5650 PRINT "FRIDAY ";
5660 GOTO 5700
5670 PRINT "SATURDAY ";
5680 GOTO 5700
5690 PRINT "SUNDAY ";
5700 IF D3>124 THEN 5740
5710 D3=D3-93
5720 PRINT "JULY "D3J" 1847"
5730 GOTO 5920
5740 IF D3>155 THEN 5780
5750 D3=D3-124
5760 PRINT "AUGUST "D3J" 1847"
5770 GOTO 5920
5780 IF D3>185 THEN 5820
5790 D3=D3-155
5800 PRINT "SEPTEMBER "D3J" 1847"
5810 GOTO 5920
5820 IF D3>216 THEN 5860
5830 D3=D3-185
5840 PRINT "OCTOBER "D3J" 1847"
5850 GOTO 5920
5860 IF D3>246 THEN 5900
5870 D3=D3-216
5880 PRINT "NOVEMBER "D3J" 1847"
5890 GOTO 5920
5900 D3=D3-246
5910 PRINT "DECEMBER "D3J" 1847"
5920 PRINT
5930 PRINT "FOOD","BULLETS","CLOTHING","MISC. SUPP.","CASH"
5940 IF B>0 THEN 5960
5950 LET B=0
5960 IF C>0 THEN 5980
5970 LET C=0
5980 IF M1>0 THEN 6000
5990 LET M1=0
6000 IF T>0 THEN 6020
6010 LET T=0
6020 IF F>0 THEN 6040
6030 LET F=0
6040 PRINT INT(F),INT(B),INT(C),INT(M1),INT(T)
6050 PRINT
6060 PRINT TAB(11);"PRESIDENT JAMES K. POLK SENDS YOU HIS"
6070 PRINT TAB(17);"HEARTIEST CONGRATULATIONS"
6080 PRINT
6090 PRINT TAB(11);"AND WISHES YOU A PROSPEROUS LIFE AHEAD"
6100 PRINT
6110 PRINT TAB(22);"AT YOUR NEW HOME"
6120 STOP
6130 REM ***SHOOTING SUB-ROUTINE***
6131 REM THE METHOD OF TIMING THE SHOOTING (LINES 6210-6240)
6132 REM WILL VARY FROM SYSTEM TO SYSTEM.  FOR EXAMPLE, H-P
6133 REM USERS WILL PROBABLY PREFER TO USE THE 'ENTER' STATEMENT.
6134 REM IF TIMING ON THE USER'S SYSTEM IS HIGHLY SUSCEPTIBLE
6135 REM TO SYSTEM RESPONSE TIME, THE FORMULA IN LINE 6240 CAN
6136 REM BE TAILORED TO ACCOMODATE THIS BY EITHER INCREASING
6137 REM OR DECREASING THE 'SHOOTING' TIME RECORDED BY THE SYSTEM.
6140 DIM S$(5)
6150 S$(1)="BANG"
6160 S$(2)="BLAM"
6170 S$(3)="POW"
6180 S$(4)="WHAM"
6190 S6=INT(RND(-1)*4+1)
6200 PRINT "TYPE "; S$(S6)
6210 B3 = CLK(0)
6220 INPUT C$
6230 B1 = CLK(0)
6240 B1=((B1-B3)*3600)-(D9-1)
6250 PRINT
6255 IF B1>0 THEN 6260
6257 B1=0
6260 IF C$=S$(S6) THEN 6280
6270 B1=9
6280 RETURN
6290 REM ***ILLNESS SUB-ROUTINE***
6300 IF 100*RND(-1)<10+35*(E-1) THEN 6370
6310 IF 100*RND(-1)<100-(40/4**(E-1)) THEN 6410
6320 PRINT "SERIOUS ILLNESS---"
6330 PRINT "YOU MUST STOP FOR MEDICAL ATTENTION"
6340 M1=M1-10
6350 S4=1
6360 GOTO 6440
6370 PRINT "MILD ILLNESS---MEDICINE USED"
6380 M=M-5
6390 M1=M1-2
6400 GOTO 6440
6410 PRINT "BAD ILLNESS---MEDICINE USED"
6420 M=M-5
6430 M1=M1-5
6440 IF M1<0 THEN 5110
6450 IF L1=1 THEN 4940
6460 GOTO 4710
6470 REM ***IDENDIFICATION OF VARIABLES IN THE PROGRAM***
6480 REM A = AMOUNT SPENT ON ANIMALS
6490 REM B = AMOUNT SPENT ON AMMUNITION
6500 REM B1 = ACTUAL RESPONSE TIME FOR INPUTTING "BANG"
6510 REM B3 = CLOCK TIME AT START OF INPUTTING "BANG"
6520 REM C = AMOUNT SPENT ON CLOTHING
6530 REM C1 = FLAG FOR INSUFFICIENT CLOTHING IN COLD WEATHER
6540 REM C$ = YES/NO RESPONSE TO QUESTIONS
6550 REM D1 = COUNTER IN GENERATING EVENTS
6560 REM D3 = TURN NUMBER FOR SETTING DATE
6570 REM D4 = CURRENT DATE
6580 REM D9 = CHOICE OF SHOOTING EXPERTISE LEVEL
6590 REM E = CHOICE OF EATING
6600 REM F = AMOUNT SPENT ON FOOD
6610 REM F1 = FLAG FOR CLEARING SOUTH PASS
6620 REM F2 = FLAG FOR CLEARING BLUE MOUNTAINS
6630 REM F9 = FRACTION OF 2 WEEKS TRAVELED ON FINAL TURN
6640 REM K8 = FLAG FOR INJURY
6650 REM L1 = FLAG FOR BLIZZARD
6660 REM M = TOTAL MILEAGE WHOLE TRIP
6670 REM M1 = AMOUNT SPENT ON MISCELLANEOUS SUPPLIES
6680 REM M2 = TOTAL MILEAGE UP THROUGH PREVIOUS TURN
6690 REM M9 = FLAG FOR CLEARING SOUTH PASS IN SETTING MILEAGE
6700 REM P = AMOUNT SPENT ON ITEMS AT FORT
6710 REM R1 = RANDOM NUMBER IN CHOOSING EVENTS
6720 REM S4 = FLAG FOR ILLNESS
6730 REM S5 = ""HOSTILITY OF RIDERS"" FACTOR
6740 REM S6 = SHOOTING WORD SELECTOR
6750 REM S$ = VARIATIONS OF SHOOTING WORD
6760 REM T = CASH LEFT OVER AFTER INITIAL PURCHASES
6770 REM T1 = CHOICE OF TACTICS WHEN ATTACKED
6780 REM X = CHOICE OF ACTION FOR EACH TURN
6790 REM X1 = FLAG FOR FORT OPTION
6800 END
```

Lost & Forgotten Island

Bruce Hicks

Lost and Forgotten Island is a survival game involving cooperation and decisions between different values. It's played by one, two or three people.

You're all stranded on a desert island. On each turn, you can individually work on the boat (to escape from a coming storm) or mine for gold. Each player also has tools to help accomplish the different tasks; tools can be traded with other players. At the end of the game you are told if you made it off the island and how much gold you have.

This program was passed along to us by Ken Modesitt of Texas Instruments and converted to Microsoft BASIC by Jeffrey Yuan. □

Bruce Hicks, School of Secondary Education, University of Illinois, Urbana, IL, 61801.

```
RUN
WELCOME TO THE LOST AND FORGOTTEN ISLAND.
WOULD YOU LIKE SOME INSTRUCTIONS? YES
LOST AND FORGOTTEN ISLAND IS A SURVIVAL GAME BASED ON
COOPERATION.  IT CONTAINS A MIXTURE OF LIFE'S VALUES.
IMAGINE:
        YOU HAVE BEEN SHIPWRECKED ON A REMOTE ISLAND.
YOU HAVE THE CHOICE OF DIGGING FOR GOLD AND/OR BUILDING
A SHIP TO SURVIVE THE APPROACHING HURRICANE.
CAN YOU SURVIVE?  IF SO, WITH HOW MUCH GOLD?

           G O O D   L U C K


HOW MANY PEOPLE (1,2, OR 3) ARE PLAYING? 2

PLAYER  1  WHAT NAME ARE YOU USING ? EMERSON

PLAYER  2  WHAT NAME ARE YOU USING ? HAWTHORNE

THIS IS DAY  1


EMERSON HAS  0  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  5 ,  0  WORK POINTS, WHICH
IS  0  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       PICKAXE
       AXE
       LUMBER
       PICKAXE

JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?


HAWTHORNE HAS  0  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  11 ,  0  WORK POINTS, WHICH
IS  0  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       PICKAXE
       CHISEL
       HAMMER
       CHISEL

JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?


THE SUM OF EVERYONE'S WORK POINTS IS  0 .

DO ANY OF YOU WISH TO TRADE TOOLS? YES

WHO (ONE NAME ONLY PLEASE) WISHES TO TRADE? EMERSON

WHO ELSE WISHES TO TRADE? HAWTHORNE

EMERSON, ARE YOU GIVING ANY GOLD IN THIS TRADE? NO

EMERSON, ARE YOU GIVING A(NY) TOOL(S) IN THIS TRADE? YES

EMERSON, HOW MANY TOOLS ARE YOU GIVING? 1

EMERSON, WHAT IS THE NAME OF A TOOL THAT YOU ARE GIVING IN TRADE? PICKAXE

HAWTHORNE, ARE YOU GIVING ANY GOLD IN THIS TRADE? NO

HAWTHORNE, ARE YOU GIVING A(NY) TOOL(S) IN THIS TRADE? YES

HAWTHORNE, HOW MANY TOOLS ARE YOU GIVING? 1

HAWTHORNE, WHAT IS THE NAME OF A TOOL THAT YOU ARE GIVING IN TRADE? CHISEL

THIS IS YOUR LAST CHANCE TO CALL OFF THE TRADE.  IF YOU
WANT TO CALL IT OFF TYPE 'X' OTHERWISE TYPE ANY OTHER LETTER AFTER
THE QUESTION MARK.
? J

DO ANY TWO OF YOU WISH TO TRADE NOW? NO

EMERSON, WHAT ARE YOU GOING TO WORK ON TODAY? BOAT

EMERSON HAS EARNED  2  MORE WORK POINTS.

EMERSON HAS BEEN INJURED BY THE PICKAXE.  HIS(HER)
TOOL PROFICIENCY WILL NOW BE CUT IN HALF.

HAWTHORNE, WHAT ARE YOU GOING TO WORK ON TODAY? BOAT

HAWTHORNE HAS EARNED  3  MORE WORK POINTS.

HAWTHORNE HAS BEEN INJURED BY THE PICKAXE.  HIS(HER)
TOOL PROFICIENCY WILL NOW BE CUT IN HALF.

THIS IS DAY  2


EMERSON HAS  0  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  3 ,  2  WORK POINTS, WHICH
IS  36  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       CHISEL
       AXE
       LUMBER
       PICKAXE

JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?


HAWTHORNE HAS  0  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  6 ,  3  WORK POINTS, WHICH
IS  64  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       PICKAXE
       HAMMER
       CHISEL
       PICKAXE
```

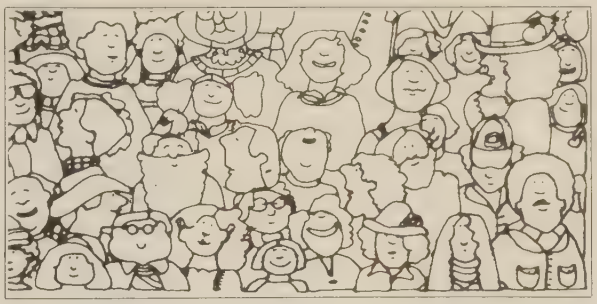
```
JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?

THE SUM OF EVERYONE'S WORK POINTS IS  5 .

THE STORM IS ABOUT TO HIT

DO ANY OF YOU WISH TO TRADE TOOLS? NO

EMERSON, WHAT ARE YOU GOING TO WORK ON TODAY? BOAT

EMERSON HAS EARNED  1  MORE WORK POINTS.

EMERSON HAS BEEN INJURED BY THE AXE.  HIS(HER)
TOOL PROFICIENCY WILL NOW BE CUT IN HALF.

HAWTHORNE, WHAT ARE YOU GOING TO WORK ON TODAY? GOLD

EMERSON HAS JUST MADE  2477  DOLLARS MORE GOLD.
HAWTHORNE HAS JUST MADE  4955  DOLLARS MORE GOLD.

THIS IS DAY  3

EMERSON HAS  2477  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  2 ,  3  WORK POINTS, WHICH
IS  48  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       CHISEL
       AXE
       LUMBER
       PICKAXE

JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?

HAWTHORNE HAS  4955  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  6 ,  3  WORK POINTS, WHICH
IS  52  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       PICKAXE
       HAMMER
       CHISEL
       PICKAXE

JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?

THE SUM OF EVERYONE'S WORK POINTS IS  7 .

THE STORM IS ABOUT TO HIT

DO ANY OF YOU WISH TO TRADE TOOLS? NO

EMERSON, WHAT ARE YOU GOING TO WORK ON TODAY? BOAT

EMERSON HAS EARNED  0  MORE WORK POINTS.

HAWTHORNE, WHAT ARE YOU GOING TO WORK ON TODAY? BOAT

HAWTHORNE HAS EARNED  2  MORE WORK POINTS.

THIS IS DAY  4

EMERSON HAS  2477  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  2 ,  4  WORK POINTS, WHICH
IS  42  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       CHISEL
       AXE
       LUMBER
       PICKAXE

JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?

HAWTHORNE HAS  4955  DOLLARS WORTH OF GOLD, A TOOL
PROFICIENCY OF  6 ,  5  WORK POINTS, WHICH
IS  58  PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:

       PICKAXE
       HAMMER
       CHISEL
       PICKAXE

JUST HIT RETURN WHEN YOU ARE READY TO GO ON.?

THE SUM OF EVERYONE'S WORK POINTS IS  9 .

THE RESULTS FOR EMERSON:

EMERSON, YOU MADE IT BACK BUT THE BOAT NEARLY SWAMPED.
SO, HALF OF YOUR GOLD WAS THROWN OVERBOARD.
THIS MEANS YOU HAVE  1238  DOLLARS WORTH OF GOLD LEFT.

THE RESULTS FOR HAWTHORNE:

HAWTHORNE, YOU MADE IT BACK TO HONOLULU BUT A
LARGE WAVE WASHED YOUR GOLD OVERBOARD.  SORRY.

DO YOU WISH TO PLAY ANOTHER GAME? NO

++++++++++++++++++++++++++++++++++++++++++++

Ok
```

```
10 DIM T$(9),T2(3,9),T3(3,9),V1(3,9),V2(3,9),N$(3),C1(3)
20 DIM V3(3,9),V4(3,9),C3(3),I$(2),J$(9),W(3),G1(3),G2(3)
30 DIM N3(2),B$(2)
40 REM ***** INSTRUCTIONS *****
50 REM ADAPTED FROM "COMPUTERS AND SOCIETY' VOL. 7-NO.3, FALL,1976
60 REM
70 PRINT"WELCOME TO THE LOST AND FORGOTTEN ISLAND."
80 PRINT"WOULD YOU LIKE SOME INSTRUCTIONS";
90 INPUT Z$
100 IF Z$="YES" THEN 140
110 IF Z$="NO" THEN  240
120 PRINT "INVALID ANSWER, PLEASE RETYPE, YES OR NO"
130 GOTO90
140 PRINT"LOST AND FORGOTTEN ISLAND IS A SURVIVAL GAME BASED ON"
150 PRINT"COOPERATION.  IT CONTAINS A MIXTURE OF LIFE'S VALUES."
160 PRINT"IMAGINE:"
170 PRINT"          YOU HAVE BEEN SHIPWRECKED ON A REMOTE ISLAND."
180 PRINT"YOU HAVE THE CHOICE OF DIGGING FOR GOLD AND/OR BUILDING"
190 PRINT"A SHIP TO SURVIVE THE APPROACHING HURRICANE."
200 PRINT"CAN YOU SURVIVE?  IF SO, WITH HOW MUCH GOLD?"
210 PRINT
220 PRINT"            G O O D   L U C K            "
230 PRINT
240 T8=RND(1)
250 FOR I=1TO 3
260 FOR J=1TO 9
270 T2(I,J)=0
280 T3(I,J)=0
290 V1(I,J)=0
300 V2(I,J)=0
310 NEXT J
320 NEXT I
330 FOR I=1 TO 2
340 N3(I)=0
350 NEXT I
360 FOR I=1 TO 3
370 W(I)=0
380 G1(I)=0
390 G2(I)=0
400 C3(I)=0
410 C1(I)=0
420 NEXT I
430 N$(3)="STORAGE"
440 FOR I=1 TO 9
450 READ T$(I)
460 NEXT I
470 PRINT
480 REM ***** NUMBER OF PLAYERS *****
490 REM
500 PRINT"HOW MANY PEOPLE (1,2, OR 3) ARE PLAYING";
510  FOR U1=1 TO 3
520 FOR U2=1 TO 9
530 T2(U1,U2)=0
540 NEXT U2
550 NEXT U1
560 INPUT N1
```

```
570 PRINT
580 IF N1<1 THEN 600
590 IF N1<=3 THEN 710
600 PRINT"YOU MUST PLAY WITH 1,2, OR 3 PLAYERS."
610 PRINT
620 GOTO 500
630 E=0
640 FOR H=1 TO N1
650 G1(H)=0
660 C3(H)=0
670 C1(H)=0
680 W(H)=0
690 NEXT H
700 C2=0
710 FOR I=1 TO N1
720 PRINT"PLAYER ";I;" WHAT NAME ARE YOU USING ";
730 INPUTN$(I)
740 PRINT
750 FOR J=1 TO 3
760 IF J=I THEN 820
770 IF N$(I)<>N$(J) THEN 820
780 PRINT"SOMEONE ELSE ALREADY HAS THIS NAME SO PLEASE CHOOSE";
790 PRINT" ANOTHER."
800 PRINT
810 GOTO 730
820 NEXT J
830 NEXT I
840 FOR U3=1 TO 3
850 FOR U4=1 TO 9
860 T2(U3,U4)=1
870 T2(U3,U4)=T2(U3,U4)*(-10)
880 NEXT U4
890 NEXT U3
900 REM
910 FOR I=1 TO 5
920 T3(1,I)=1
930 NEXT I
940 T3(1,6)=2
950 T3(2,7)=1
960 T3(2,8)=1
970 T3(2,9)=2
980 E=6-N1
990 FOR I=1 TO N1
1000  FOR J=1 TO E
1010 R1=INT(RND(1)*9+1)
1020 T2(I,J)=R1
1030 V1(I,J)=T3(1,R1)
1040 V2(I,J)=T3(2,R1)
1050 NEXT J
1060 C1(I)=INT(RND(1)*11+2)
1070 NEXT I
1080 GOSUB 1130
1090 DATA"AXE","CHISEL","HAMMER","NAILS AND SCREWS","SAW"
1100 DATA"LUMBER","SHOVEL","PICKAXE","EXPLOSIVES"
1110 STOP
1120 REM
1130 REM ***** SUBROUTINE LAFIS21 *****
1140 REM
1150 REM THIS IS LAFIS 21
1160 REM ***** TRADING TOOLS *****
1170 IF N1=1 THEN 1270
1180 IF N1=3 THEN 1200
1190 N$(3)="STORAGE"
1200 J$(1)="A"
1210 J$(2)="ANOTHER"
1220 FOR I=3 TO 9
1230 J$(I)=J$(I-1)
1240 NEXT I
1250 I$(1)="WHO (ONE NAME ONLY PLEASE) WISHES TO TRADE"
1260 I$(2)="WHO ELSE WISHES TO TRADE"
1270 S=4
1280 FOR M=1 TO 5
1290 REM ***** WHICH DAY? *****
1300 PRINT"THIS IS DAY ";M
1310 PRINT
1320 GOSUB 4060
1330 PRINT
1340 PRINT
1350 IF S=1 THEN 4040
1360 IF S>3 THEN 1420
1370 S=S-1
1380 PRINT"THE STORM IS ABOUT TO HIT"
1390 PRINT
1400 IF N1=1 THEN 2970
1410 GOTO 1460
1420 IF M=5 THEN 1450
1430 X=INT(RND(1)*4+1)
1440 IF X<>4 THEN 1460
1450 S=3
1460 IF N1=1 THEN 2970
1470 PRINT"DO ANY OF YOU WISH TO TRADE TOOLS";
1480 INPUT A$
1490 PRINT
1500 IF A$="YES" THEN 1550
1510 IF A$="NO"THEN 2970
1520 PRINT"PLEASE TRY AGAIN.  YOU MUST ANSWER `YES' OR `NO'."
1530 PRINT
1540 GOTO 1460
1550 FOR I=1 TO 3
1560 FOR J=1 TO 9
1570 T3(I,J)=T2(I,J)
1580 V3(I,J)=V1(I,J)
1590 V4(I,J)=V2(I,J)
1600 NEXT J
1610 G2(I)=G1(I)
1620 NEXT I
1630 FORJ=1 TO 2
1640 N3(J)=1
1650 NEXT J
1660 FOR I=1 TO 2
1670 PRINTI$(I);
1680 INPUTB$(I)
1690 PRINT
1700 IF B$(I)=N$(1) THEN 1790
1710 IF B$(I)=N$(2)THEN 1780
1720 IF B$(I)=N$(3) THEN 1770
1730 PRINT"YOU MUST ANSWER WITH `";N$(1);"', `";N$(2);"', OR `";N$(3)
1740 PRINT"--------PLEASE TRY AGAIN."
1750 PRINT
1760 GOTO 1670
1770 N3(I)=N3(I)+1
1780 N3(I)=N3(I)+1
1790 NEXT I
1800 FOR I=1 TO 2
1810 N4=1
1820 PRINTB$(I);", ARE YOU GIVING ANY GOLD IN THIS TRADE";
1830 INPUTC$
1840 PRINT
1850 IF C$="X"THEN 2890
1860 IFC$="NO" THEN 2110
1870 IF C$="YES" THEN 1980
1880 IF C$="T" THEN 1940
1890 PRINT"PLEASE TRY AGAIN.  YOU MUST ANSWER `YES', `NO'," 
1900 PRINT"`X' (TO CALL OFF THE TRADE), OR `T' (TO SEE THE LIST OF "
1910 PRINT"TOOLS WHICH EVERYONE HAD BEFORE THE START OF THIS TRADE)."
1920 PRINT
1930 GOTO 1820
1940 PRINT"YOUR SITUATION AT THIS TIME"
1950 PRINT
1960 GOSUB4110
1970 GOTO 1820
 980 PRINTB$(I);", HOW MUCH GOLD (IN DOLLARS) ARE YOU GOING TO GIVE";
1990 INPUT A7
2000 PRINT
2010 IF G1(N3(I))>=A7 THEN 2050
2020 PRINT"YOU MAY NOT GIVE MORE THAN YOU HAVE (";G1(N3(I));" DOLLARS"
2030 PRINT
2040 GOTO 1980
2050 IF A7>=0 THEN 2090
2060 PRINT"YOU MAY NOT INPUT A NEGATIVE NUMBER.  TRY AGAIN."
2070 PRINT
2080 GOTO 1980
2090 G2(N3(I))=G1(N3(I))-A7
2100 G2(N3(3-I))=G1(N3(3-I))+A7
2110 PRINTB$(I);", ARE YOU GIVING A(NY) TOOL(S) IN THIS TRADE";
2120 INPUT D$
2130 PRINT
2140 IF D$="X" THEN 2890
2150 IF D$="NO"THEN 2730
2160 IF D$="YES" THEN 2270
2170 IF D$="T" THEN 2230
2180 PRINT"TRY AGAIN.  YOU MUST ANSWER `YES', `NO', `T' (TO"
2190 PRINT"SEE THE TOOLS EVERYONE HAD BEFORE THIS TRADE"
2200 PRINT"STARTED), OR `X' (TO CALL OFF THE TRADE)."
2210 PRINT
2220 GOTO 2110
2230 PRINT"YOUR SITUATION AT THIS TIME"
2240 PRINT
2250 GOSUB 4110
2260 GOTO 2110
2270 PRINTB$(I);", HOW MANY TOOLS ARE YOU GIVING";
2280 INPUT N5
2290 PRINT
2300 FOR J=1 TO N5
2310 PRINTB$(I);", WHAT IS THE NAME OF ";J$(J);" TOOL THAT ";
2320 PRINT"YOU ARE GIVING IN TRADE";
2330 INPUT E$
2340 PRINT
2350 IF E$="X"THEN 2890
2360 IF E$="B"THEN 2730
```

```
2370 IF E$="1"THEN 2580
2380 FOR R=1 TO 9
2390 IF T$(R)=E$THEN 2480
2400 NEXT R
2410 PRINT"PLEASE USE THE NAME OF A TOOL, USE `B' IF YOU WANT TO"
2420 PRINT"GO AHEAD WITH THE TRADE WITHOUT GIVING MORE TOOLS,"
2430 PRINT"USE `T' IF YOU WANT TO SEE THE LIST OF TOOLS EVERYONE"
2440 PRINT"HAD BEFORE THIS TRADE STARTED,"
2450 PRINT"OR USE `X' IF YOU WANT TO CALL OFF THE TRADE."
2460 PRINT
2470 GOTO 2310
2480 FOR K=1 TO 9
2490 IF T2(N3(I),K)=RTHEN 2620
2500 NEXT K
2510 PRINTB$(I);", YOU DO NOT HAVE THIS TOOL.  PLEASE TRY AGAIN."
2520 PRINT"YOU MUST USE THE NAME OF A TOOL YOU HAVE, USE `B' TO GO"
2530 PRINT"AHEAD WITH THE TRADE WITHOUT GIVING MORE TOOLS, USE `X'"
2540 PRINT"CALL OFF THE TRADE, OR USE `T' TO SEE THE LIST OF TOOLS WHICH
 EVERYONE"
2550 PRINT"HAD BEFORE THE START OF THIS TRADE."
2560 PRINT
2570 GOTO 2310
2580 PRINT"YOUR SITUATION AT THIS TIME"
2590 PRINT
2600 GOSUB 4110
2610 GOTO 2310
2620 T3(N3(I),K)=-10
2630 V3(N3(I),K)=0
2640 V4(N3(I),K)=0
2650 FOR L=N4 TO 9
2660 IF T3(N3(3-I),L)=-10 THEN 2680
2670 NEXT L
2680 T3(N3(3-I),L)=R
2690 V3(N3(3-I),L)=V1(N3(I),K)
2700 V4(N3(3-I),L)=V2(N3(I),K)
2710 N4=L+1
2720 NEXT J
2730 NEXT I
2740 PRINT"THIS IS YOUR LAST CHANCE TO CALL OFF THE TRADE.  IF YOU"
2750 PRINT"WANT TO CALL IT OFF TYPE `X' OTHERWISE TYPE ANY OTHER LETTER
AFTER"
2760 PRINT"THE QUESTION MARK."
2770 INPUT G$
2780 PRINT
2790 IF G$="X"THEN 2890
2800 FOR I=1 TO3
2810 FOR J=1 TO 9
2820 T2(I,J)=T3(I,J)
2830 V1(I,J)=V3(I,J)
2840 V2(I,J)=V4(I,J)
2850 NEXT J
2860 B1(I)=B2(I)
2870 NEXT I
2880 REM
2890 PRINT"DO ANY TWO OF YOU WISH TO TRADE NOW";
2900 INPUT H$
2910 PRINT
2920 IF H$="YES"THEN 1550
2930 IF H$="NO"THEN 2970
2940 PRINT"PLEASE TRY AGAIN.  YOU MUST ANSWER `YES' OR`NO'."
2950 PRINT
2960 GOTO 2890
2970 FOR I=1 TO N1
2980 IF C3(I)=1 THEN 4000
2990 P1=C1(I)
3000 REM ***** WHAT TYPE OF WORK TODAY? *****
3010 PRINTN$(I);", WHAT ARE YOU GOING TO WORK ON TODAY";
3020 INPUT A$
3030 PRINT
3040 IF A$="BOAT" THEN 3430
3050 IF A$="GOLD" THEN 3110
3060 PRINT "PLEASE ANSWER `BOAT' IF YOU WANT TO WORK ON THE BOAT"
3070 PRINT"OR `GOLD' IF YOU WANT TO MINE GOLD."
3080 PRINT
3090 GOTO 3010
3100 REM ***** WORKING ON SOME GOLD *****
3110 FOR J=1 TO 9
3120 IF T2(I,J)<>1 THEN 3280
3130 PRINTN$(I);", DO YOU WISH TO USE THE AXE TO MINE GOLD?"
3140 PRINT"REMEMBER THAT THE AXE DROPS GREATLY IN VALUE"
3150 PRINT"IF IT IS USED TO MINE GOLD."
3160 INPUT C$
3170 PRINT
3180 IF C$="NO" THEN 3280
3190 IF C$="YES"THEN 3230
3200 PRINT"PLEASE TRY AGAIN. YOU MUST USE  YES' OR `NO'."
3210 PRINT
3220 GOTO 3130
3230 V2(I,J)=V1(I,J)
3240 P1=P1+V2(I,J)*C1(I)
3250 V1(I,J)=(INT((V1(I,J)/2)*10^.5))/10
3260 V2(I,J)=0
3270 GOTO 3290
3280 P1=P1+V2(I,J)*C1(I)
3290 NEXT J
3300 Z8=RND(1)*2+1
3310 FOR K=1 TON1
3320 IF C3(K)=1 THEN 3400
3330 C4=0
3340 IF K<>I THEN 3360
3350 C4=.25
3360 C4=C4+.25
3370 Z9=C4*Z8*200+P1
3380 PRINTN$(K);" HAS JUST MADE ";INT(Z9);" DOLLARS MORE GOLD."
3390 G1(K)=G1(K)+Z9
3400 NEXT K
3410 PRINT
3420 GOTO 3590
3430 REM ***** WORKING ON THE BOAT *****
3440 B(I)=1
3450 FOR J=1 TO 9
3460 IF T2(I,J)<>2 THEN 3520
3470 FOR K=1 TO 9
3480 IF T2(I,K)<>3 THEN 3510
3490 P1=P1 +2*V1(I,J)*C1(I)
3500 GOTO 3530
3510 NEXT K
3520 P1=P1+V1(I,J)*C1(I)
3530 NEXT J
3540 Z8=P1/12
3550 PRINTN$(I);" HAS EARNED ";INT(Z8);" MORE WORK POINTS."
3560 PRINT
3570 W(I)=W(I)+Z8
3580 C2=C2+Z8
3590 Y=INT(RND(1)*(9+C1(I))+1)
3600 FOR J=1 TO 9
3610 IF T2(I,J)=1 THEN 3640
3620 IF T2(I,J)=3 THEN 3640
3630 IF T2(I,J)<>8 THEN 3700
3640 IF Y<>T2(I,J)THEN 3990
3650 PRINTN$(I);" HAS BEEN INJURED BY THE ";T$(T2(I,J));".  HIS(HER)"
3660 PRINT"TOOL PROFICIENCY WILL NOW BE CUT IN HALF."
3670 PRINT
3680 C1(I)=INT((C1(I)/2)+.5)
3690 J=9
3700 IF T2(I,J)<>9 THEN 3990
3710 IF Y<>T2(I,J) THEN 3990
3720 PRINTN$(I);" HAS BEEN KILLED BY THE ACCIDENTAL"
3730 PRINT"DISCHARGE OF SOME OF THE EXPLOSIVES.  PLEASE"
3740 PRINT"NOTIFY HIS(HER) FRIENDS AND RELATIVES IF YOU MAKE IT BACK."
3750 PRINT
3760 C3(I)=1
3770 IF D7=1 THEN 3800
3780 N$(I)="STORAGE"
3790 D7=1
3800 G1(I)=0
3810 G2(I)=0
3820 IF N1=1 THEN 4030
3830 FOR K=1 TO 9
3840 IF T2(I,K)=-10 THEN 3970
3850  R3=INT(RND(1)*N1+1)
3860 IF R3=I THEN 3850
3870 FOR L=1 TO 9
3880 IF T2(R3,L)<>-10 THEN 3960
3890 T2(R3,L)=T2(I,K)
3900 V1(R3,L)=V1(I,K)
3910 V2(R3,L)=V2(I,K)
3920 T2(I,K)=-10
3930 V1(I,K)=0
3940 V2(I,K)=0
3950 L=9
3960 NEXT L
3970 NEXT K
3980 J=9
3990 NEXT J
4000 NEXT I
4010 NEXT M
4020 GOSUB 4060
4030 REM
4040 GOSUB 4440
4050 REM THE FOLLOWING IS THE SUBROUTINE STATE
4060 FOR I=1 TO 3
4070 FOR J=1 TO 9
4080 T3(I,J)=T2(I,J)
4090 NEXT J
4100 NEXT I
4110 C4=C2
4120 FOR M=1 TO N1
4130 PRINT
4140 PRINT
```

```
4150 IF C3(H)=1 THEN 4340
4160 IF C2<>0 THEN 4180
4170 C2=1
4180 E=INT((W(H)/C2)*100+.5)
4190 PRINTN$(H);" HAS ";INT(G1(H));" DOLLARS WORTH OF GOLD, A TOOL"
4200 PRINT"PROFICIENCY OF ";C1(H);", ";INT(W(H));" WORK POINTS, WHICH"
4210 PRINT"IS ";E;" PERCENT OF THE TOTAL, AND THE FOLLOWING TOOLS:"
4220 PRINT
4230 FOR J2=1 TO 9
4240 K$=" "
4250 IF T2(H,J2)=-10 THEN 4290
4260 IF T2(H,J2)=T3(H,J2) THEN 4280
4270 K$="*"
4280 PRINTTAB(6);K$;T$(T2(H,J2))
4290 NEXT J2
4300 PRINT
4310 PRINT"JUST HIT RETURN WHEN YOU ARE READY TO GO ON.";
4320 INPUT F$
4330 PRINT
4340 NEXT H
4350 PRINT
4360 PRINT"THE SUM OF EVERYONE'S WORK POINTS IS ";INT(C4);"."
4370 PRINT
4380 C2=C4
4390 RETURN
4400 DATA"AXE","CHISEL","HAMMER","NAILS AND SCREWS","SAW"
4410 DATA"LUMBER","SHOVEL","PICKAXE","EXPLOSIVES"
4420 RETURN
4430 REM
4440 REM ***** SUBROUTINE LAFIS31 *****
4450 REM
4460 DEF FNE(Z7)=2.718^Z7
4470 DEF FNC(Z8)=FNE(Z8)+FNE(-Z8)
4480 DEF FNS(Z9)=FNE(Z9)-FNE(-Z9)
4490 C4=C2
4500 IF C2<>0 THEN 4520
4510 C2=1
4520 IF Z1=3 THEN 4590
4530 C4=0
4540 FOR I=1 TO N1
4550 IF C3(I)=1 THEN 4570
4560 IF W(I)/C2<.25 THEN 4580
4570 C4=C4+W(I)
4580 NEXT I
4590 FOR I=1 TO N1
4600 IF C3(I)=1 THEN5030
4610 PRINT
4620 PRINT
4630 REM
4640 REM ***** THE RESULTS *****
4650 REM
4660 PRINT
4670 PRINT"THE RESULTS FOR ";N$(I);":"
4680 PRINT
4690 PRINT
4700 IF W(I)/C2<.25 THEN 4720
4710 W(I)=C4
4720 Z1=INT(60*2.718^((-W(I))/6))
4730 Z2=INT(50*(1+((FNS((7-W(I))/8.5))/FNC((7-W(I))/8.5))))
4740 Z3=INT(50*(1+(FNS((14-W(I))/5)/FNC((14-W(I))/5))))
4750 R5=INT(RND(1)*101)
4760 IF B(I)<>1THEN 4960
4770 IF R5>Z1 THEN 4820
4780 PRINT"PROPER CONDOLENCES WILL BE SENT TO THE FRIENDS"
4790 PRINT"AND RELATIVES OF ";N$(I);" WHO DROWNED DURING"
4800 PRINT"TYPHOON URSULA."
4810 GOTO 5030
4820 IF R5>Z2 THEN 4860
4830 PRINTN$(I);", YOU MADE IT BACK TO HONOLULU BUT A"
4840 PRINT"LARGE WAVE WASHED YOUR GOLD OVERBOARD.  SORRY."
4850 GOTO 5030
4860 IF R5>Z3 THEN 4920
4870 PRINTN$(I);", YOU MADE IT BACK BUT THE BOAT NEARLY SWAMPED."
4880 PRINT"SO, HALF OF YOUR GOLD WAS THROWN OVERBOARD."
4890 PRINT"THIS MEANS YOU HAVE ";INT(G1(I)/2);
4900 PRINT" DOLLARS WORTH OF GOLD LEFT."
4910 GOTO 5030
4920 PRINTN$(I);", CONGRATULATIONS----------"
4930 PRINT"YOU MADE IT WITH ALL YOUR GOLD,";INT(G1(I));
4940 PRINT" DOLLARS WORTH."
4950 GOTO 5030
4960 IF R5>=97 THEN 5000
4970 PRINTN$(I);" DID NOT GET OFF THE ISLAND AND WAS "
4980 PRINT"KILLED BY TYPHOON URSULA."
4990 GOTO 5030
5000 PRINTN$(I);", YOU SURVIVED TYPHOON URSULA, BUT LOST ALL YOUR GOLD"
5010 PRINT"AND HAD BETTER START MAKING SMOKE SIGNALS BECAUSE YOU WERE"
5020 PRINT"LEFT BEHIND."
5030 NEXTI
5040 PRINT
5050 PRINT"DO YOU WISH TO PLAY ANOTHER GAME";
5060 INPUT C$
5070 PRINT
5080 PRINT
5090 PRINT
5100 PRINT
5110 PRINT"************************************************"
5120 PRINT
5130 IF C$="NO" THEN 5190
5140 IF C$="YES" THEN 70
5150 PRINT"YOU MUST ANSWER `YES' OR `NO'.  PLEASE TRY AGAIN."
5160 PRINT
5170 GOTO 5050
5180 RETURN
5190 END
Ok
```



Computers and the Population Problem

David H. Ahl

Population growth has an immense effect on the use of various resources on the earth and in individual countries. It also leads to interesting questions like, "If the population growth of the United States and Mexico continue at their present rates, when will the population of Mexico exceed that of the United States?"

In any event, it's probably worthwhile to update the old population problems and simulations every once in a while with new figures. Below are some representative 1977 figures. If you would like a complete 1977 vs. 1950 population chart showing every country of the world, write to The Environmental Fund, 1302 Eighteenth St., NW, Washington, DC 20036. They're a non-profit agency, so you might include $1 or $2 to help defray the cost — the chart is well worth it. (We've found the population estimates of The Environmental Fund to be better documented and, therefore, probably more accurate, than those of both the United Nations and the U.S. Census Bureau.)

POPULATION DOUBLING TIME

Population Growth Rate %	Number of Years to double Population
0.5	139
1.0	69
1.5	46
2.0	35
2.5	28
3.0	23
3.5	20
4.0	17

Major Area	*Population Estimates (Millions) Mid 1977*	*Growth Rate (%)*	*Birth Rate (Per 1000 Population)*	*Death Rate (Per 1000 Population)*	*1950 Population (Millions)*
World	4307.4	2.1	34	13	2542.8
Africa	451.2	2.8	46	20	218.9
Asia	2505.7	2.4	38	14	1408.5
North America	249.0*	1.5	14	9	166.1
Latin America	341.8	2.8	37	9	164.4
Europe	478.8	0.6	15	11	392.4
USSR	259.4	1.0	18	9	180.1
Oceania	21.8	1.8	22	9	12.4
Selected Countries					
Libya	2.6	4.1	45	15	.95
Nigeria	89.3	2.9	49	23	34 3
Iraq	11.8	3.4	48	15	5.2
Kuwait	1.1	6.1	47	5	.1
Japan	113.5	1.1	17	6	83.8
China, People's Rep.	987.3	2.4	36	12	556.3
United States	225.6*	1.5	14	9	152.3
Mexico	62.6	3.5	44	7	26.5
Brazil	116.1	2.9	37	9	53.3
Sweden	8.3	0.4	12	11	7.0
Germany, Fed. Rep.	62.2	0.1	10	12	50.0
German, Dem. Rep.	16.8	-0.2	11	14	18.4

*U.S. population figures include an estimate of 1.7 million from illegal immigration, derived from U S Immigration and Naturalization Service reports. Corresponding adjustments have *not* been made to other figures, most notably Mexico

World 2

James L. Murphy

Here's your chance to create a better world, by experimenting with a simulation model.

In [1] Jay W. Forrester presented a world population model interrelating many variables including population, natural resources, food supply, pollution, capital investment, and quality of life. The computer program presented there was written in the Dynamo language. This is a version of the same model written in BASIC. The program can best be used in conjunction with the model description given in *World Dynamics*.

In lines 5000-6000 the variables as used in *World Dynamics* are related to the corresponding variables of the BASIC program along with the line number in which they are defined. This makes it quite easy to make modifications in the program, allowing the user to duplicate the results reported by Forrester as well as to experiment on his own to try for a "better world" according to your own values and assumptions.

Figure 1 shows the outcome when using the assumptions of the original model. In this model the world runs down for lack of natural resources. You can test what would happen if you assumed the natural resources available in 1900 were double what was originally assumed available (see Figure 2). In line 5460 of the program, you see that natural resources initial is N1 and defined in line 2270 which was changed in the run of Figure 2 from 900E9 to 1800E9. This caused pollution to grow out of control, so one might want to introduce a pollution-control program. In line 5610, we see that pollution normal (after switch time) is S4 and defined in line 2400. By changing S4 from one to one-half, we see in Figure 3 what happens if, starting in 1970, the amount of pollution produced per person is somehow cut in half.

To simulate a birth-control program, we see in line 5050 that B2 is the birth-rate normal (after switch time) and is defined in line 2110. Figure 4 shows the result of setting the birth rate equal to the death rate beginning in 1970.

Many other changes can be made in this same way by using the descriptions in lines 5000-6000 to locate where a particular variable is to be defined in the program, and then modifying that line to suit your own assumptions regarding that particular variable.

There are three different sets of five variables each whose graphs may be produced. The variable A9 in line 1680 determines which of the three sets of variables is to be graphed (see Figure 5). If you would rather have a list of numbers for the variables rather than see their graphs, simply change line 2500 as in Figure 6.

There are many assumptions made in *World Dynamics* as to the relationship between various variables within the model. You may change these assumptions by changing the corresponding DATA statements in lines 1240 to 1670. For example, to change the way in which you believe the death rate is affected by pollution, you would alter the values in line 1510. It is fairly easy to do this if you first look at page 41 of [1] where the assumptions made by Forrester regarding these two variables are given. With these few examples and the program listing, I think you will now be able to create your own future worlds.

*Dept. of Mathematics, California State College, 5500 State College Pkwy, San Bernardino, CA 92407.

[1] Forrester, Jay W. *World Dynamics*. Wright-Allen Press, Inc., Cambridge, Massachusetts, 1971.

```
1000 !*****************************************************************
1010 !
1020 !A PROGRAM WRITTEN BY JAMES L. MURPHY
1030 !AT CALIF. STATE COLLEGE SAN BERNARDINO
1040 !BASED ON A MODEL BY JAY W. FORRESTER
1050 !IN "WORLD DYNAMICS", WRIGHT-ALLEN PRESS 1971.
1060 !
1070 !*****************************************************************
1080 !
1090 !A DESCRIPTION OF VARIABLES BEGINS AT LINE 5000
1100 !
1110 DIM Y(21), B(21,10)
1120 DIM P(51), C(5)
1140 PRINT "WORLD POPULATION MODEL 9-15-77"
1160 MAT B = ZER
1170 MAT Y = ZER
1180 FOR I = 0 TO 21
1190 FOR J = 0 TO 10
1200 READ B(I,J)
1210 NEXT J
1220 NEXT I
1230 RESTORE
1240 REM BRCM FROM CR
1250 DATA 1.05, 1, .9,.7,.6,.55,0,0,0,0,0
1260 REM DRCM FROM CR
1270 DATA .9,1,1.2,1.5,1.9,3,0,0,0,0,0
1280 REM FCM FROM CR
1290 DATA 2.4,1,.6,.4,.3,.2,0,0,0,0,0
1300 REM QLC FROM CR
1310 DATA 2,1.3,1,.75,.55,.45,.38,.3,.25,.22,.2
1320 REM FPCI FROM CIR4
1330 DATA .5,1,1.4,1.7,1.9,2.05,2.2,0,0,0,0
1340 REM POLCM FROM CIR
1350 DATA .05,1,3,5.4,7.4,8,0,0,0,0,0
1360 REM NREM FROM NRFR
1370 DATA 0,.15,.5,.85,1,0,0,0,0,0,0
1380 REM QLM FROM MSL
1390 DATA .2,1,1.7,2.3,2.7,2.9,0,0,0,0,0
1400 REM NRMM FROM MSL
1410 DATA 0,1,1.8,2.4,2.9,3.3,3.6,3.8,3.9,3.95,4
1420 REM CIM FROM MSL
1430 DATA .1,1,1.8,2.4,2.8,3,0,0,0,0,0
1440 REM DRMM FROM MSL
1450 DATA 3,1.8,1,.8,.7,.6,.53,.5,.5,.5,.5
1460 REM BRMM FROM MSL
1470 DATA 1.2,1,.85,.75,.7,.7,0,0,0,0,0
1480 REM FPM FROM POLR
1490 DATA 1.02,.9,.65,.35,.2,.1,.05,0,0,0,0
1500 REM DRPM FROM POLR
1510 DATA .92,1.3,2,3.2,4.8,6.8,9.2,0,0,0,0
1520 REM BRPM FROM POLR
1530 DATA 1.02,.9,.7,.4,.25,.15,.1,0,0,0,0
1540 REM QLP FROM POLR
1550 DATA 1.04,.85,.6,.3,.15,.05,.02,0,0,0,0
1560 REM BRFM FROM FR
1570 DATA 0,1,1.6,1.9,2,0,0,0,0,0,0
1580 REM DRFM FROM FR
1590 DATA 30,3, 2,1.4,1,.7,.6,.5,.5,0,0
1600 REM CFIFR FROM FR
1610 DATA 1,.6,.3,.15,.1,0,0,0,0,0,0
1620 REM QLF FROM FR
1630 DATA 0,1,1.8,2.4,2.7,0,0,0,0,0,0
1640 REM CIQR FROM QLM/QLF
1650 DATA .7,.8,1,1.5,2,0,0,0,0,0,0
1660 REM POLAT FROM POLR
1670 DATA .6,2.5,5,8,11.5,15.5,20,0,0,0,0
1680 A9 = 1
1690 C$ = "PNSQC"
1700 IF A9 = 1 THEN 1740
1710 C$ = "MRIWA"
1720 IF A9 = 2 THEN 1740
1730 C$ = "GFLBX"
1740 CHANGE C$ TO C
1750 B$(1) = "P2 - POPULATION"
1760 B$(2) = "N2 - NATURAL RESOURCES"
1770 B$(3) = "S7 - POLLUTION RATIO"
1780 B$(4) = "Q2 - QUALITY OF LIFE"
1790 B$(5) = "C2 - CAPITAL INVESTMENT"
1800 IF A9 = 1 THEN 1930
1810 B$(1) = "F8-MATERIAL STAND.OF LIV."
1820 B$(2) = "N6-NAT.RESOURCE USAGE"
1830 B$(3) = "G4-CAP.INVEST.RATIO"
1840 B$(4) = "G5-CAP.INVEST.RAT.AGRIC."
1850 B$(5) = "C4-CAP.INVEST.AG.FRAC."
1860 IF A9 = 2 THEN 1990
1870 B$(1) = "S6-POLLUTION GENERATED"
1880 B$(2) = "F6-FOOD RATIO"
1890 B$(3) = "P2/R3-LIFE EXPECTANCY"
1900 B$(4) = "B3/P2-BIRTH RATE"
1910 B$(5) = "G3-CROWDING RATIO"
1920 GO TO 2050
1930 W(1) = 8E9
1940 W(2) = 2E12
1950 W(3) = 40
1960 W(4) = 2
1970 W(5) = 2E10
1980 IF A9 = 1 THEN 2100
1990 W(1) = 4
2000 W(2) = 20E9
2010 W(3) = 4
2020 W(4) = 4
2030 W(5) = 1
2040 IF A9 = 2 THEN 2100
2050 W(1) = 40E9
2060 W(2) = 2
2070 W(3) = 100
2080 W(4) = .1
2090 W(5) = 2
2100 B1 = .026
2110 B2 = .026
2120 C1 = .4E9
2130 C2 = C1
2140 C3 = .2
2150 C4 = C3
2160 C5 = 15
2170 C6 = .3
2180 C7 = .025
2190 C8 = .025
2200 F1 = 1
2210 F3 = 1
2220 F4 = 1
2230 F5 = 1
2240 F7 = 135E6
2250 G1 = .05
2260 G2 = .05
2270 N1 = 900E9
2280 N2 = N1
2290 N3 = 1
2300 N4 = 1
2310 P1 = 1.65E9
2320 P2 = P1
2330 P3 = 26.5
2340 Q1 = 1
2350 R1 = .014
2360 R2 = .014
2370 S1 = .2E9
2380 S2 = S1
2390 S3 = 1
2400 S4 = 1
2410 S5 = 3.6E9
2420 T1 = 1900
2430 T2 = 2100
2440 T3 = 1970
2450 T4 = 1
2460 T5 = T1
2470 L3 = 5
2480 M3 = 5
2490 M2 = -1
2500 W1 = T2 + 1
2510 W2 = 20
2520 PRINT
2720 IF W1 < T2 THEN 2780
2730 FOR I = 1 TO 5
2740PRINT MID(C$,I,1);" REPRESENTS ";B$(I);"--- SCALED TO";W(I)
2750 NEXT I
2760 PRINT
2770P$="       0.2       0.4       0.6       0.8       1.0 "
2775PRINT"      0.0";P$
2780 IF T5<T3 THEN 2860
2790 B1 = B2
2800 R1 = R2
2810 F4 = F5
2820 S3 = S4
2830 C7 = C8
2840 G1 = G2
2850 N3 = N4
2860 G3 = P2/(F7*P3)
2870 I = -1
2880 X = G3
2890 K = 5
2900 GOSUB 3860
2910 GOSUB 3860
```

Grain vs. People

An article in *The Wall Street Journal* (January 20, 1975) reports that world food production is growing. It is. USDA supplies the following figures on grain production:

1970 . . . 1.11 billion metric tons
1974 . . . 1.12 " " "

Thus, grain production in 1974 was 10 million metric tons higher than in 1970.

BUT, in the same 5 year period, the world produced 355.1 million additional people, whose grain requirements each year (at the world per capita average of 311 kilos), are not 10 million more metric tons, but 110 million more metric tons.

Some have said the earth can hold many more people than it now has. From the standpoint of room alone this is so but do we want our voyage on earth to be as one on a cruise ship or one on a troop ship?

—Jack J. Combes, "Some Thoughts on Limits to Growth," *NAM Reports*, Aug. 13, 1973, p. 9.

```
2920 GOSUB 3860
2930 X = 2*G3
2940 K = 10
2950 GOSUB 3860
2960 G4 = C2/P2
2970 G5 = G4*C4/C6
2980 X = G5
2990 K = 6
3000 GOSUB 3860
3010 X = G4
3020 K = 5
3030 GOSUB 3860
3040 S6 = P2*S3*Y(5)
3050 M5 = N2/N1
3060 X = 4*M5
3070 K = 4
3080 GOSUB 3860
3090 F2 = G4*(1-C4)*Y(6)/(1-C6)
3100 F8 = F2 /F1
3110 X = F8
3120 K = 5
3130 GOSUB 3860
3140 K = 10
3150 GOSUB 3860
3160 K = 5
3170 GOSUB 3860
3180 X = 2*F8
3190 K = 10
3200 GOSUB 3860
3210 X = F8
3220 K = 5
3230 GOSUB 3860
3240 M6 = P2*M3*Y(8)
3250 G6 = P2*Y(9)*G1
3260 G7 = C2*C7
3270 S7 = S2/S5
3280 X = S7/10
3290 K = 6
3300 GOSUB 3860
3310 GOSUB 3860
3320 GOSUB 3860
3330 GOSUB 3860
3340 F6 = Y(4)*Y(2)*Y(12)*F4/F3
3350 X = F6
3360 K = 4
3370 GOSUB 3860
3380 X = 4*F6
3390 K = 8
3400 GOSUB 3860
3410 R3 = P2*R1*Y(10)*Y(13)*Y(17)*Y(1)
3420 B3 = P2*B1*Y(16)*Y(11)*Y(0)*Y(14)
3430 X = 2*F6
3440 K=4
3450 GOSUB 3860
3460 X = F6
3470 GOSUB 3860
3480 X = 2*Y(7)/Y(19)
3490 GOSUB 3860
3500 X = S7/10
3510 K = 6
3520 GO SUB 3860
3530 S8 = S2/Y(21)
3540 Q2=Q1*Y(7)*Y(3)*Y(19)*Y(15)
3550 IF T5 < W1 THEN 3590
3560 IF T5 = W1 THEN 4510
3570 IF ((T5 -W1) -W2*INT((T5-W1)/W2)) = 0 THEN 4550
3580 GO TO 3600
3590 GOSUB 3980
3600 P2 = P2 + T4*(B3 - R3)
3610 N2 = N2 - T4*N6
3620 S2 = S2 + T4*(S6 - S8)
3624 IF S2 > 0 THEN 3630
3626 S2 = 0
3630 C4 = C4 + (T4/C5)*(Y(18)*Y(20)-C4)
3640 C2 = C2 + T4*(G6 - G7)
3650 T5 = T5 + T4
3660 IF T5 < (T2 + T4) THEN 2780
3670 PRINT
3680 PRINT
3690 PRINT
3700GOTO4600
3850 REM **********TABLE INTERPOLATION ROUTINE**********
3860 I = I + 1
3870 IFX<0THEN X=0
3880 J = INT(X)+1
3890 IF X<K THEN 3950
3930 Y(I) = D(I,K)
3940 GO TO 3960
3950 Y(I) = D(I,J) + (X-J)*(D(I,J)-D(I,J-1))
3960 RETURN
3970 REM **********PLOT ROUTINE**********
3980 ON A9 GOTO 3985,4040,4100
3985 V(1)=P2
3990 V(2) = N2
4000 V(3) = S7
4010 V(4) = Q2
4020 V(5) = C2
4030 GOTO4150
4040 V(1) = F8
4050 V(2) = N6
4060 V(3) = G4
4070 V(4) = G5
4080 V(5) = C4
4090 GOTO4150
4100 V(1) = S6
4110 V(2) = F6
4120 V(3) = P2/R3
4130 V(4) = B3/P2
4140 V(5) = G3
4150 L6 = 0
4160 L2 = T5 - T1
4170 L4 = L2 - INT(L2/L3)*L3
4180 IF L4 <> 0 THEN 4500
4190MAT P=CON:MAT=(32)*P
4280 M2 = M2+1
4290 IF (M2 - INT(M2/M3)*M3) <> 0 THEN 4340
4300 MAT P = CON
4310 MAT P =(45)*P
4330 L6 = 1
4340 FORL1=1TO51STEP10:P(L1)=124
4345NEXT L1:FOR L1=1TO5
4350 L4 = INT(50*V(L1)/W(L1)+.5) +1
4360 IF L4 > 51 THEN L4=51
4370 IF L4 < 1 THEN L4=1
4380 P(L4) = C(L1)
4390 NEXT L1
4400 P(0) = 51
4410 CHANGE P TO P$
4460 IF L6 = 0 THEN 4490
4470 PRINT T5;" ";P$
4480 RETURN
4490 PRINT "       ";P$
4500 RETURN
4510 PRINT "DATE","POP","NR","CI","CIAF",
4520 PRINT "POLR", "QL","CR","CIR","CIRA",
4530 PRINT "MSL","FR","NRFR","BR/P","LIFE EXP"
4540 PRINT
4550 PRINT T5,P2,N2,C2,C4,
4560 PRINT S7,Q2,G3,G4,G5,
4570 PRINT F8,F6,N5,B3/P2,P2/R3
4580 PRINT
4590 GO TO 3600
4600END

Ready
```

> *"Stated most simply, if mankind is to live in the state of material well-being that technology can make possible, then, given the finite size and resources of the planet, there are just too many of us already."*
>
> —Philip Handler, *President*, National Academy of Sciences. *Bioscience*, July 1975, p. 425.

```
5000 '3420 B3    BR BIRTH RATE (PEOPLE/YEAR)
5010 '2900 Y(0)  BRCM BIRTH-RATE-FROM-CROWDING MULTIPLIER
5020 '3370 Y(16) BRFM BIRTH-RATE FROM-FOOD MULTIPLIER
5030 '3230 Y(11) BRMM BIRTH-RATE FROM-MATERIAL MULTIPLIER
5040 '2100 B1    BRN BIRTH RATE NORMAL (FRACTION/YEAR)
5050 '2110 B2    BRN1  "    "    "  (AFTER SWITCH TIME)
5060 '3320 Y(14) BRPM BIRTH-RATE-FROM-POLLUTION MULTIPLIER
5070 '3450 Y(18) CFIFR CAPITAL FRACTION INDICATED BY FOOD RATIO
5080 '3640 C2    CI CAPITAL INVESTMENT
5090 '3630 C4    CIAF CAPITAL-INVESTMENT-IN-AGRICULTURE FRACTION
5100 '2140 C3    CIAFI  "       "       "       "       "  INITIAL
5110 '2170 C6    CIAFN  "       "       "       "       "  NORMAL
5120 '2160 C5    CIAFT  "       "       "       "      "ADJUST TIME
5130 '3260 G7    CID CAPITAL INVESTMENT DISCARD
5140 '2180 C7    CIDN   "       "       "  NORMAL
5150 '2190 C8    CIDN1  "       "       "   "  (AFTER SWITCH TIME)
5160 '3250 G6    CIG CAPITAL INVESTMENT GENERATION
5170 '2250 G1    CIGN   "       "       "  NORMAL
5180 '2260 G2    CIGN1  "       "       "   "  (AFTER SWITCH TIME)
5190 '2120 C1    CII CAPITAL INVESTMENT INITIAL
5200 '3170 Y(9)  CIM CAPITAL INVESTMENT MULTIPLIER
```

```
5210 '3490 Y(20) CIQR   "        "   FROM QUALITY RATIO
5220 '2960 G4    CIR CAPITAL INVESTMENT RATIO (CAPITAL UNITS/PERSON)
5230 '2970 G5    CIRA   "        "   IN AGRICULTURE
5240 '2860 G3    CR CROWDING RATIO
5250 '3410 R3    DR DEATH RATE (PEOPLE/YEAR)
5260 '2910 Y(1)  DRCM DEATH-RATE-FROM-CROWDING MULTIPLIER
5270 '3400 Y(17) DRFM DEATH-RATE-FROM-FOOD MULTIPLIER
5280 '3200 Y(10) DRMM DEATH-RATE-FROM-MATERIAL MULTIPLIER
5290 '2350 R1    DRN DEATH RATE NORMAL (FRACTION/YEAR)
5300 '2360 R2    DRN1  "    "    "  (AFTER SWITCH TIME)
5310 '3310 Y(13) DRPM DEATH-RATE-FROM-POLLUTION MULTIPLIER
5320 '3090 F2    ECIR EFFECTIVE-CAPITAL-INVESTMENT RATIO (CAP/PERSON)
5330 '2200 F1    ECIRN   "        "       "  NORMAL
5340 '2220 F4    FC FOOD COEFFICIENT
5350 '2230 F5    FC1 FOOD COEFFICIENT  (AFTER SWITCH TIME)
5360 '2920 Y(2)  FCM FOOD FROM CROWDING MULTIPLIER
5370 '2210 F3    FN FOOD NORMAL (FOOD UNITS/PERSON/YEAR)
5380 '3000 Y(4)  FPCI FOOD POTENTIAL FROM CAPITAL INVESTMENT
5390 '3300 Y(12) FPM FOOD FROM POLLUTION MULTIPLIER
5400 '3340 F6    FR FOOD RATIO
5410 '2240 F7    LA LAND AREA  (SQUARE KILOMETERS)
5420 '3100 F8    MSL MATERIAL STANDARD OF LIVING
5430 '3610 N2    NR NATURAL RESOURCES
5440 '3080 Y(6)  NREM NATURAL RESOURCES EXTRACTION MULTIPLIER
5450 '3050 N5    NRFR   "        "   FRACTION REMAINING
5460 '2270 N1    NRI    "        "   INITIAL
5470 '3150 Y(8)  NRMM NATURAL RESOURCES FROM MATERIAL MULTIPLIER
5480 '2290 N3    NRUN   "        "   USAGE NORMAL (UNITS/PERSON/YEAR)
5490 '2300 N4    NRUN1  "        "   "  (AFTER SWITCH TIME)
5500 '3240 N6    NRUR NATURAL RESOURCES USAGE RATE (UNITS/YEAR)
5510 '3600 P2    P POPULATION  (PEOPLE)
5520 '2330 P3    PDN POPULATION DENSITY NORMAL (PEOPLE/SQ.KILOMETER)
5530 '2310 P1    PI POPULATION INITIAL
```

```
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11
```

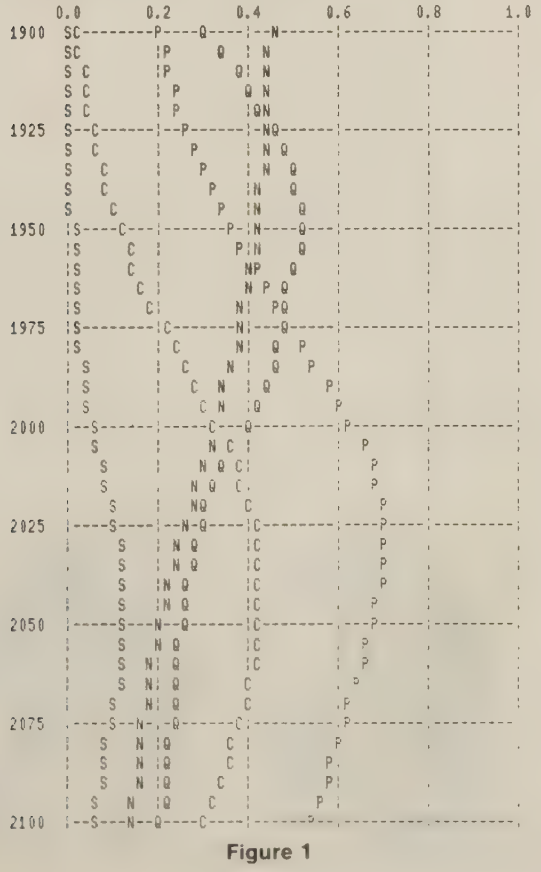


Figure 1

```
5540 '3620 S2    POL POLLUTION  (POLLUTION UNITS)
5550 '3530 S8    POLA POLLUTION ABSORPTION (POL.UNITS/YEAR)
5560 '3520 Y(21) POLAT   "        "   TIME  (YEARS)
5570 '3030 Y(5)  POLCM POLLUTION FROM CAPITAL MULTIPLIER
5580 '3040 S6    POLG POLLUTION GENERATION (UNITS/YEAR)
5590 '2370 S1    POLI   "    INITIAL
5600 '2390 S3    POLN POLLUTION NORMAL
5610 '2400 S4    POLN1  "        "   (AFTER SWITCH TIME)
5620 '3270 S7    POLR   "    RATIO
5630 '2410 S5    POLS POLLUTION STANDARD (POL. UNITS)
5640 '3540 Q2    QL QUALITY OF LIFE  (SATISFACTION UNITS)
5650 '2950 Y(3)  QLC QUALITY OF LIFE FROM CROWDING
5660 '3470 Y(19) QLF QUALITY OF LIFE FROM FOOD
5670 '3130 Y(7)  QLM QUALITY OF LIFE FROM MATERIAL
5680 '3330 Y(15) QLP QUALITY OF LIFE FROM POLLUTION
5690 '2340 Q1    QLS QUALITY OF LIFE STANDARD
5700 '2420 T1    INITIAL TIME (IN CALENDER YEARS)
5710 '2430 T2    ENDING TIME
5720 '2440 T3    SWITCHING TIME
5730 '2450 T4    CHANGE IN TIME  (ONE YEAR)
5740 '2460 T5    TIME
5750 '3980-4140 V(I)  I-TH VARIABLE FOR PLOT
5760 '1930-2090 W(I)  I-TH VARIABLE SCALE
5770 '2470 L3    FREQUENCY OF PLOT
5780 '2480 M3    FREQUENCY OF DATES IN PLOT
5790 '1680 A9    WHICH OF THREE GRAPHS  1  2  OR  3
5800 '2500 W1    MINIMUM PRINT TIME
5810 '2510 W2    PRINT FREQUENCY
6000 END
READY.
```

```
2270 N1 = 1800E9 ! TO SEE WHAT HAPPENS IF WE DOUBLE NATURAL
RESOURCES AVAILABLE.IN YEAR 1900
RUNNH
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11
```

Figure 2

```
2270 N1 = 1800E9

2400 S4 = .5 ! SIMULATE POLUTION CONTROL PROGRAM BY REDUCING
POLUTION PRODCED PER PERSON TO 1/2 WHAT IT WOULD BE BEFORE 1970.

Ready

RUNNH
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11
```

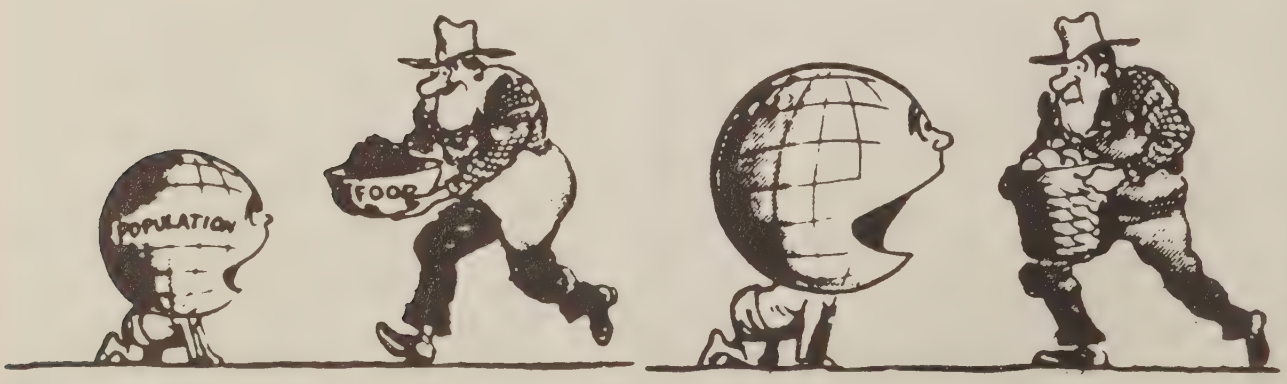
```
      0.0       0.2       0.4       0.6       0.8       1.0
1900  SC--------P----Q----,---------,---------¦----N----¦
      SC         ,P      Q  ,        ¦         ¦    N    ,
      S C        ,P        Q,        ¦         ¦    N    ,
      S C        , P       Q         ,         ¦    N    ,
      S C        , P       , Q       ,         ¦    N    ¦
1925  S--C-------,--P-------,--Q------¦---------¦----N----¦
      S  C       ,   P     ,   Q     ,         ¦    N    ,
      S    C     ,     P   ,     Q   ,         ¦    N    ,
      S    C     ¦       P ¦       Q ¦         ¦    N    ¦
      S     C    ,         P¦       Q¦         ¦    N    ¦
1950  ¦S----C----¦---------P-¦------Q--¦---------¦---N-----¦
      ,S      C  ,           P¦      Q¦         ¦    N    ¦
      ,S       C ¦            ,P     Q¦         ¦    N    ¦
      ,S        C¦            , P    Q¦         ¦    N    ¦
      ,S         C            ,  P   Q¦         ¦    N    ¦
1975  ¦S---------¦C---------¦----Q----¦---------¦N--------¦
      ,S         , C       ¦     Q P ¦         ¦N        ¦
      ,S         ,   C     ¦     Q   P         N         ¦
      ¦S         ,     C   ¦   Q       P       N         ¦
      ,S         ,       C , Q         ¦P      N¦        ¦
2000  ¦S---------¦--------C-¦Q---------¦---P----N-¦---------¦
      ,S         ,         CQ          ,     P N ¦         ¦
      , S        ¦          QC         ¦      PN ¦         ¦
      , S        ¦           Q,  C     ¦       N P¦        ¦
      , S        ¦         Q ,    C    ¦      N   P¦       ¦
2025  ¦-S--------¦-------Q---¦------C---¦----N------P---------¦
        S        ¦       Q  ¦        C  ¦  N       ¦P        ¦
      , S        ¦     Q    ¦          C¦ N       ¦ P       ¦
      , S        ¦    Q     ¦           CN        ¦  P      ¦
      ¦ S        ¦    Q     ¦           NC        ¦   P     ¦
2050  ,---S------¦---Q-------¦---------N,-C-------¦----P----¦
      ¦   S      ¦  Q       ¦          N ¦ C      ¦     P   ¦
      ¦   S      ¦ Q        ¦         N  ¦  C     ¦     P   ¦
      ¦   S      ¦ Q        ¦        N   ¦  C     ¦     P   ¦
      ¦   S      ¦ Q        ¦        N   ¦  C     ¦     P   ¦
2075   ---S------,-Q--------¦-----N-----¦---C------¦-----P---¦
      ¦   S      ¦Q         ¦    N      ¦   C      ¦     P   ¦
      ¦   S      ¦Q         ¦  N        ¦   C      ¦     P   ¦
      ¦   S      ¦Q         ¦ N         ¦   C      ¦   P     ¦
      ¦   S      ¦Q         ¦ N         ¦ C        ¦   P     ¦
2100  ¦---S------¦Q---------¦N---------¦-C--------¦-P-------¦
```

Figure 3

```
2110 B2 = .014  ! SIMULATE BIRTH CONTROL PROGRAM BY SETTING BASIC BIRTH
RATE EQUAL TO DEATH RATE STARTING IN SWITCH YEAR 1970.

Ready

RUNNH
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11
```

```
      0.0       0.2       0.4       0.6       0.8       1.0
1900  SC--------P----Q----,--N------¦---------¦---------¦
      SC         ,P      Q  , N     ,         ,         ,
      S C        ,P        Q¦ N     ,         ,         ,
      S C        , P       Q N      ,         ,         ,
      S C        , P       ,QN      ,         ,         ,
1925  S--C-------,--P-------¦-NQ-----¦---------¦---------¦
      S  C       ,   P     , N Q    ,         ,         ,
      S    C     ¦     P   ¦ N  Q   ,         ,         ,
      S    C     ,       P ,N   Q   ,         ,         ,
      S     C    ¦         P¦N    Q  ,         ,         ,
1950  ¦S----C----¦---------P-¦N----Q--¦---------¦---------¦
      ,S      C  ¦           P¦N     Q¦         ¦         ¦
      ,S       C ,            NP    Q ,         ,         ,
      ¦S        C¦            N P Q    ¦         ¦         ¦
      ¦S         C            N¦  PQ   ¦         ¦         ¦
1975  ¦S---------¦C--------N¦--P---Q--¦---------¦---------¦
      ¦ S        ¦ C       N¦  P      Q¦         ¦         ¦
      ¦ S        ¦   C     N ¦  P     Q ¦         ¦         ¦
      ¦ S        ¦     C N   ¦  P     ¦ Q       ¦         ¦
      ¦  S       ¦       C N ¦  P     ¦  Q      ¦         ¦
2000  ¦--S-------¦-------C---¦----P----¦---Q-----¦---------¦
      ¦   S      ¦        NC ,    P   ¦ Q       ¦         ¦
      ¦    S     ¦      N  C ,      P ,Q       ¦         ¦
      ,     S    ¦     N    C       P  Q       ,         ,
      ¦      S   ¦    N     C        P Q¦       ¦         ¦
2025  ,-------S---,---N-------C---------Q---------,---------,
      ¦       S  ¦ N         ¦C      QP ¦         ¦         ¦
      ¦        S ¦ N         ¦ C Q   P  ¦         ¦         ¦
      ¦         S¦N          ¦  CQ   P  ¦         ¦         ¦
      ¦         S¦N          ¦  C    P  ¦         ¦         ¦
2050  ¦---------SN----------,QC----P--¦---------¦---------¦
      ¦          SN         QC     P  ¦         ¦         ¦
      .           S         QC     P  ,         ,         ,
      ¦         SN¦         QC     P  ¦         ¦         ¦
      ,          S¦         QC     P  ,         ,         ,
2075  ,-------SN--¦--------QC¦---P------¦---------¦---------¦
      ¦      S N  ¦        C ¦  P      ¦         ¦         ¦
      ,       S N ,         QC    P    ,         ,         ,
      ¦     S N   ¦       C   ¦  P     ¦         ¦         ¦
      ,      S N  ,        C   ,  P     ,         ,         ,
2100  ,---S--N---¦----CQ---,P---------,---------¦---------¦
```

Figure 4



```
1800 PO = 3     TO GRAPH A DIFFERENT SET OF FIVE VARIABLES 1,2, OR 3

READY

RUNNH
WORLD POPULATION MODEL 9-15-77

1 REPRESENTS BIRTH-DEATH GENERATED--- SCALED TO .4E 11
2 REPRESENTS FOOD RATIO--- SCALED TO 2
3 REPRESENTS LIFE EXPECTANCY --- SCALED TO 100
4 REPRESENTS POPULATION RATE--- SCALED TO .4
  REPRESENTS INDUSTRIAL RATIO--- SCALED TO 2

      0.0        0.2        0.4        0.6        0.8        1.0
1900  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
1925  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
1950  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
1975  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
2000  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
2025  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
2050  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
2075  .1---------2---------3---------4---------5---------6---------.
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
      .1         2         3         4         5         6         .
2100  .1---------2---------3---------4---------5---------6---------.
```

Figure 5

```
2500 W1 = 1900  ! TO PRINT VALUES RATHER THAN GRAPHS

READY

RUNNH
WORLD POPULATION MODEL 9-15-77

DATE          POP           NR            CI            CIAF
POLR          QL            CR            CIR           CIRA
NSL           FR            NRFR          BR/P          LIFE EXP

 1900         .165E 10      .9E 12        .4E 9         .2
 .555556E-1   .611596       .461216       .242424       .161616
 .277056      1.03861       1             .318735E-1    36.7659

 1920         .196116E 10   .88628E 12    .889143E 9    .21521
 .161558      .859049       .548192       .453377       .325238
 .503645      1.10128       .984756       .315628E-1    51.2495

 1940         .253342E 10   .859044E 12   .171298E 10   .22909
 .314741      1.01171       .708153       .676155       .516334
 .724319      1.08527       .954494       .297325E-1    61.0276

 1960         .328022E 10   .811839E 12   .298576E 10   .258357
 .579815      1.00748       .916902       .910233       .788883
 .907703      1.00869       .902044       .270773E-1    67.5253

 1980         .41307E 10    .740856E 12   .466868E 10   .295855
 1.1492       .938059       1.15463       1.13024       1.11463
 1.01631      .987254       .823174       .253664E-1    70.9236

 2000         .502579E 10   .646852E 12   .658197E 10   .321481
 2.25276      .781284       1.40483       1.30964       1.40341
 1.02345      .966456       .718724       .238568E-1    62.9535

 2020         .556544E 10   .544239E 12   .805598E 10   .331894
 3.74293      .606776       1.55568       1.4475        1.60139
 .893431      .940777       .604777       .230056E-1    57.2273

 2040         .553121E 10   .451767E 12   .858396E 10   .322477
 4.90884      .530151       1.54611       1.55191       1.66819
 .755169      .951912       .501964       .235889E-1    53.6276

 2060         .520575E 10   .3775862 12   .8225329E 10  .294769
 4.77267      .490361       1.45514       1.50366       1.57044
 .612091      .963657       .4.354        .02364        54.4806

 2080         .482322E 10   .322621E 12   .7284861 10   .277566
 3.52549      .458373       1.3482        1.31483       1.34743
 .471009      .975234       .363683       .2444E-1      51.5 16

 2100         .439335E 10   .2844928 12   .6.8553E 10   .2.8627
 2.3275       .41795        1.22603       1.18367       1.20868
 .355871      .977047       .316411       .2 .62E-1     .. .16
```

Figure 6

Gene Basset, Scripps Howard Newspapers

A Social Science Survey Program

Dr. James Owens

This program will add a whole new dimension to social science classes . . . from elementary school to college! All you need is an inexpensive 4K microcomputer.

Dr. James Owens, Professor of Management, The American University, Washington, D.C.

Many teachers, especially high-school and college teachers, can easily use a small computer (even one with as little as 4K of RAM memory) to conduct and process professional-type social science surveys. Obviously, such surveys and studies contribute to the teacher's research, and publication-record in the "publish or perish" academic world. More importantly (and interestingly), social science surveys of a teacher's own classroom students, when quickly tabulated and processed, provide rich material for enhancing and making *relevant* course topics. For example, the function of the Social Security program in the USA can be rescued from dull abstraction in classroom lectures by a survey question forcing the student to record his own opinion and discuss it within the context of the tabulated opinions and statistical conclusions of the class population as a whole. The same is true for a multitude of other routine high-school and college course topics such as the ERA amendment, function of taxation in the USA, corporation profits, political voting patterns and more. Such classroom applications are virtually infinite in number and potential course subject areas.

The "Survey" program described here was developed and used with an Ohio Scientific 6502 computer providing 8K BASIC in ROM and 20K of RAM memory. However, the essential values of the "social science survey" in classroom use can be achieved in as little as 4K of RAM memory and most BASIC-IN-ROM systems (e.g., other Microsoft BASIC systems such as the Level II TRS-80, Apple II, Sorcerer, PET).

The Questionnaire

The program permits a highly refined graduation of questionnaire responses ranging from *very strong* agreement to *very strong* disagreement with all items on the questionnaire (including provision for "no opinion" responses). Figure 1 presents the basic format of such a questionnaire. The program here dimensions fifty rows for fifty questions or fewer. Depending on memory available, the program can process about forty full DATA lines, including 1K of memory for the RUN, within the contraints of a typical 4K RAM memory system. A 4K system, therefore, can handle a survey of, for example, a forty-person sample with twenty-five questions in the questionnaire, a 100-person sample with ten questions in the questionnaire, or any proportionate combination of sample number and question number between the two extremes exampled. Conserving memory, as this program does, any user, restricted to 4K of RAM, will still find room in memory for very useful research or classroom-oriented social science survey projects. It should be noted that a large survey, for example a 100-person sample with 50 questions in the questionnaire, will require about 20K of RAM.

The Video Display—or Output—of the Program

The program generates for each row of output-information (each row being equivalent to a particular question in the questionnaire) a full twenty-one "columns" of calculated values. Since, however, it would be a rare conventional page of paper or computer video screen that could list twenty-one horizontal columns, the programmer has the choice of arraying the columns

closely (using a ';') or vertically (using no punctuation at all in the Line 490 PRINT statement) or, as used in this program, completing the PRINT statement with a comma. Only the latter programming tactic emerges as useful in terms of esthetics, styling and *readability* — and, thus, is used here. The "comma," ending Line 490, produces for each Question/Item number an easily readable and systematically consistent array of 'columns' in five vertical rows as illustrated in Figure 2. Use of the full twenty-one 'columns' permits calculation output of professional-type values such as standard deviation for the sample total as well as internal sample elements such as "males" vs "females" or "age 30 or over" vs "age 29 or under." Most importantly, calculations of the standard error of the difference between "means" (such as between the mean for Males and the mean for Females or the mean for those over age 30 vs those under age 29) become possible — as well as reports on the "significance of the difference between means" (as in Figure 2 columns #18 and #20). Note: Many users of this program may not need, or want, all of the calculations-results produced in the RUN output; in which case, unwanted lines can easily be deleted from the program. Also, the 21-column output display formatted as in Figure 2 requires a video display of 64 characters per line available with many small systems such as Radio Shack's TRS-80, OSI's C2 Challenger series of systems and others; if your system provides fewer than 64 characters/line display, a semi-colon — or no punctuation at all — at the end of line 490 will position the twenty-one columns in a more readable format than the comma used in the program here.

Just prior to the video display of the first question (as in Figure 2), the program generates a display summarizing the sample and its subsamples, as follows:

Total Sample	= 9		
Total Males	= 4	Total Females	= 5
Total Age 30+	= 5	Total Age 29-	= 4

The Program Methodology and Documentation

An explicit goal of the program development was to economize memory use to a minimum. One main method was, of course, to eliminate all documentation (REM statements) from the program lines. That documentation will, therefore, be presented here in line-order by line number (or groups of lines).

Directions: Please enter in the column at the right a number from "1" to "9" for each question/item indicating your degree of agreement or disagreement on the 1-9 scale where:

1 = "I disagree very strongly."
2 = "I disagree strongly."
3 = "I disagree substantially."
4 = "I disagree slightly."
5 = "No Opinion or Preference"
6 = "I agree slightly."
7 = "I agree substantially."
8 = "I agree strongly."
9 = "I agree very strongly."

For Reference please check appropriate categories: Male ✓ Female
Age 30 or over ✓ Age 29 or under

#	QUESTION/ITEMS	Enter 1-9 in this Column
1.	The Social Security System, including premiums and payments, should be abolished.	1
2.	The "ERA" amendment should be ratified and become law.	2
3.	All taxes, including Federal and State income taxes, should be decreased.	8
4.	Etc.	

Figure 2. Illustration of typical video display of calculations (question #1 is illustrated). Note: In the actual video display generated by the program the column numbers and descriptions above are deleted (to conserve memory) and only the calculated values are printed. See Figure 3 for actual video display output.

Column # 1 Question #	Column # 2 Males' Score	Column # 3 Females' Score	Column # 4 Age 30+ Score	Column #5 29- Score
1	21	17	12	26
Column # 6 Mean for Total Sample	**Column # 7 Mean for Males**	**Column # 8 Mean for Females**	**Column # 9 Mean for 30+**	**Column #10 Mean for 29-**
4.22	5.25	3.4	2.4	6.5
Column # 11 Standard Deviation for Total Sample	**Column # 12 Standard Deviation for Males**	**Column # 13 Standard Deviation for Females**	**Column # 14 Standard Deviation 30+**	**Column# 15 StandardDeviation 29-**
2.44	2.68	1.85	1.5	1.12
Column # 16 Number of "No Opinion"	**Column # 17 Standard Error of the Difference between the Male vs Female Means**	**Column # 18 Significance of the M/F Difference (3 = 99% level of confidence)**	**Column # 19 Standard Error of Difference between 30+ vs 29- Means**	**Column#20 Significance of 30+/29- Difference**
2	1.58	1.17	.87	4.7
Column # 21 % of "No Opinion				
.22				

Note: In the actual video display generated by the program the column numbers and descriptions above are deleted (to conserve RAM memory) and only the calculated values in the order above are printed. See figure 3 for actual numerical video display output for Question/items #2 and 3

Line 10 rounds off all decimals to two only for purposes of readability. Without this function the video display becomes overcrowded with useless decimal expressions of six or more digits (such as 4.2247653). The "FNH" in 490 implements the function.

Line 20 dimensions an array of fifty rows, each row consisting of twenty-one columns. As mentioned above, each row (1,2,3 etc.) represents a single, particular survey question identified by the same question number and row number (thus, Row 1's displayed calculations display the calculated results for Question 1 of the survey, as in Figure 2). Remember, too, that within each row, as displayed on the video screen, the row's twenty-one columns are themselves formatted into five vertical elements ("rows" within the basic Row) and five horizontal columns (Figure 2).

Lines 30 to 80 enter the row (or question) number, from 1 to 50, into the first column of each row and set all twenty remaining columns for each row at zero.

Lines 90 to 230 place survey responses (numerical values) into Columns 2,3,4 and 5 for each question (row). The variable P reads either a 2 or a 3 (2 = Male, 3 = Female) from the first DATA entry in each DATA line (Line 901) and keeps a count of total males and females; the variable R does the same count for respondents aged 30 and over vs those aged 29 and under (30+ vs 29–) as read from the second datum from each line (every data line representing one individual respondent). The variable R reads either a 4 or a 5 (4 = age 30+, 5 = age 29–). Although obvious, it may be of interest to many readers that the P and R variables can be used, *with no structural change in the program*, to record and calculate a variety of "categories" such as smokers vs non-smokers, liberal vs conservatives, Protestants vs non-Protestants, Yankee baseball fans vs their opposite and so on. Also, if only a single discrimination of sample categories is needed, or none at all, then simply eliminate the program lines which involve the P variable and the R variable respectively. Line 95 registers dummy data to move the program toward its end. Lines 180 and 210 calculate and total the sums of squared values needed for the later standard deviation calculation and store these temporarily in Columns 17 and 20. After the standard deviation calculations are completed, and the sums of the squared values no longer relevant, Columns 17 through 20 are "erased" and used to record, and display, meaningful values as in the Figure 2 format. Although a bit intricate, or "tricky," the loops in 170 through 220 minimize memory use by quantum proportions as compared to a series of GOTO's. Line 190 records "no opinion" responses for their total as displayed in Column 16.

Lines 250 through 290 calculate the averages (arithmetic means) for, respectively, the total sample, males, females, age 30+ and age 29–, while line 300 records the sample total. (See Figure 2.)

Lines 310 through 350 calculate the standard deviations for, respectively, the total sample, males, females, age 30+ and age 29– (see any standard Statistics text for statement and explanation of the formula for the standard deviation). Results are displayed in Columns #11 through #15, respectively, as in Figure 2.

Lines 360 and 370 calculate the *standard error of the difference between the Means* for males vs females and age 30+ vs 29–, respectively, recording these values for printing in Columns 17 and 19 for each question/item (see Figure 2). (Formula is available in any standard Statistics text.)

```
TOTAL SAMPLE =  9
TOTAL MALES =  4                TOTAL FEMALES =  5
TOTAL AGE 30+ = 5               TOTAL 29- = 4
=============================================================
 1              21              17              12              26
 4. 22          5. 25           3. 4            2. 4            6. 5
 2. 44          2. 68           1. 85           1. 5            1. 12
 2              1. 58           1. 17           . 87            4. 7
 0

*************************************************************
 2              12              36              27              21
 5. 33          3               7. 2            5. 4            5. 25
 2. 62          1. 58           1. 6            2. 42           2. 86
 2              1. 07           3. 94           1. 79           . 08
 0

*************************************************************
 3              30              35              37              28
 7. 22          7. 5            7               7. 4            7
 1. 47          1. 5            1. 41           1. 36           1. 58
 2              . 98            . 51            1               . 4
 0

*************************************************************
 4              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0

*************************************************************
 5              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0

*************************************************************
 6              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0

*************************************************************
 7              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0              0               0               0               0
 0
```

Figure 3. Actual program output (for 3 sample questions from DATA statements in Program A). Note: The underlining of 3.94 in Question 2 and .51 and .4 in Question 3 is to draw attention to an *enormous statistical significance* in the difference between "male" and "female" responses to Question 2. This, contrasted with virtually no significant differences in Question 3 ("males" and "females" of all ages indicate a preference for lower taxes).

Lines 380 and 390 calculate the *significance of the difference between means* (where 2 = a 95% statistical confidence level, 3 = a 99% statistical confidence level, etc.). For example, in the hypothetical and illustrative data here, there is virtually *no significance* in the difference between male and female responses on "abolishing the Social Security System" (Q 1) but a *very* significant difference in the responses of respondents *over and under age 30* (Column 20 for Q 1 shows a "4.7" value, meaning 4.7 standard errors of difference or a *greater than 99% probability* that the difference is *real*

rather than due to any kind of sampling chance). In Q 2, regarding the ERA amendment, Column 18 shows a similar huge significance of difference between *male* and *female* responses (3.94) while, in Q 3, no significant differences appear in either Columns 18 (males vs females) or 20 (age 30+ vs 29–), meaning that there is not much difference among the categories about the desire for tax reduction.

Lines 420 through 550 are the printing routine. Again, please note the comma ending Line 490.

In each DATA line, one for each survey respondent, the first datum indicates sex, the second datum indicates age, the third datum indicates numerical response to the first question, the fourth datum indicates response to the second question and the fifth datum indicates response to the third question. *Important Note*: In actual use, the user will want more than the three "questions" illustrated here; thus, Line 150 must be changed to establish the *exact* number of questions such as 150 FOR Q = 1 TO 10 (for a survey of ten questions) or 150 FOR Q = 1 TO 50 (for a survey of fifty questions).

The last DATA line must end with a -1 dummy data to move the program to an end.

A special note about the .0001 in lines 380 and 390: the .0001 assures a RUN and avoidance of an occasional "division by zero" error message without affecting the substantial results produced in Columns 18 and 20 for each row (question) number.

Varieties of Application

With minor changes in the basic program, a user can obtain professional-type statistical results for a large variety of combinations of sample categories, number and type of questions in a questionnaire, statistical calculations and video display.

Program A. Social Science Survey Program. (With the exception of the title, REM statements do not appear in the program in order to conserve memory. Documentation for the program statements is provided in the article. The program lines, excepting DATA, use 1.4K of memory.)

```
3  REM ------------------------------------
4  REM    SOCIAL SCIENCE SURVEY PROGRAM
5  REM        BY DR. JAMES OWENS
7  REM
8  REM    CREATIVE COMPUTING MAGAZINE
9  REM ------------------------------------
10 DEF FNH(X)=INT(X*100+.5)/100
20 DIM S(50,21)
30 FOR R=1 TO 50
40 LET S(R,1)=R
50 FOR C=2 TO 21
60 LET S(R,C)=0
70 NEXT C
80 NEXT R
90 READ P
95 IF P=-1 THEN GOTO 240
100 IF P=2 THEN LET M=M+1
110 IF P=3 THEN LET F=F+1
120 READ R
130 IF R=4 THEN LET O=O+1
140 IF R=5 THEN LET U=U+1
150 FOR Q=1 TO 3
160 READ A
170 LET S(Q,P)=S(Q,P)+A
180 LET S(Q,P+15)=S(Q,P+15)+A^2
190 IF A=5 THEN LET S(Q,16)=S(Q,16)+1
200 LET S(Q,R)=S(Q,R)+A
210 LET S(Q,R+15)=S(Q,R+15)+A^2
220 NEXT Q
230 GOTO 90
240 FOR R=1 TO 50
250 LET S(R,6)=(S(R,2)+S(R,3))/(M+F)
260 LET S(R,7)=S(R,2)/M
270 LET S(R,8)=S(R,3)/F
280 LET S(R,9)=S(R,4)/O
290 LET S(R,10)=S(R,5)/U
300 LET T=M+F
310 LET S(R,11)=SQR((S(R,17)+S(R,18))/T)-S(R,6)^2)
320 LET S(R,12)=SQR((S(R,17)/M)-S(R,7)^2)
330 LET S(R,13)=SQR((S(R,18)/F)-S(R,8)^2)
340 LET S(R,14)=SQR((S(R,19)/O)-S(R,9)^2)
350 LET S(R,15)=SQR((S(R,20)/U)-S(R,10)^2)
360 LET S(R,17)=SQR((S(R,12)^2/M)+(S(R,13)^2/F))
370 LET S(R,19)=SQR((S(R,14)^2/O)+(S(R,15)^2/U))
380 LET S(R,18)=ABS((S(R,7)-S(R,8)))/(S(R,17)+1E-04)
390 LET S(R,20)=ABS((S(R,9)-S(R,10)))/(S(R,19)+1E-04)
400 NEXT R
420 PRINT "-----------------------------------------"
430 PRINT"TOTAL SAMPLE = ";T
440 PRINT"TOTAL MALES = ";M,"TOTAL FEMALES = ";F
450 PRINT"TOTAL AGE 30+ =";O,"TOTAL 29- =";U
460 PRINT"========================================="
470 FOR R=1 TO 50
480 FOR C=1 TO 21
490 PRINT FNH(S(R,C)),
500 NEXT C
510 PRINT
520 PRINT
530 PRINT
540 PRINT"*****************************************"
550 NEXT R
560 END
901 DATA 2,4,1,2,8
902 DATA 2,4,5,4,9
903 DATA 3,4,2,9,5
904 DATA 3,4,3,7,7
905 DATA 3,4,1,5,8
906 DATA 2,5,8,1,5
907 DATA 2,5,7,5,8
908 DATA 3,5,6,9,6
909 DATA 3,5,5,6,9,-1
```

Star Merchant

Lloyd Johnson

It seems that interstellar trade takes us back to the days of the Yankee Clipper. Return with us now to those thrilling days of tomorrow...

Historical Background

Early in the 26th century, the 50XFTL drive was developed. This drive, when properly installed on a spaceship, would cause a controlled warping of space enabling the spaceship to travel at fifty times the speed of light (50XFTL). Massive colonization of the nearby stars took place in the following two centuries, due to the crowded conditions on inhabitable planets of the solar system and the development of this drive.

By the mid 29th century, large orbiting space stations (starports) were constructed at the ten most populated star systems. These starports had facilities for docking and refueling starships as well as massive cargo storage capability. The construction of these starports was closely paralleled by a simplification of starship design. With the advent of the starports, it was no longer necessary for a starship to land on a planet. This eliminated the need for atmospheric streamlining, as well as the large reaction engines required to lift the starship from the planetary surface, while it substantially increased the cargo hold of starships.

The type of cargo which will be available for purchase at any particular starport is difficult to predict, since most of the cargos did not originate at that star system, but were brought there by other merchant starships. Coordination of trade routes to guarantee cargo availability at a starport had never occurred due to the independent nature of the star merchants and the slow communication between the star systems.

As trade developed between the starports, each starport was assigned a trade classification. Although the trade classification is useless in determining which cargos might be available for purchase, it is extremely useful in predicting the price of the cargo.

Game Description

When you play Star Merchant you will find yourself commander of a merchant starship. Your goal is to make enough money by trading cargos to stay in business and regain your initial investment for the lease of the starship. There are ten different starports where trade is conducted and 36 different types of cargos which may be traded. A brief set of instructions for the game is listed in lines 90-260 of the program and is also listed in the sample run.

The different types of cargos range from agricultural produce and raw materials to industrially produced items, such as weapons and machinery. The price at which these cargos will be traded is dependent upon the trade classification of the starport where the item is being traded. For example, farm machinery might bring top dollar at a starport with an agricultural trade class, whereas the price of grain there will probably be very low.

The starport names, distances and directions listed for command 5 (LIST STARPORTS) were taken from the actual names, distances and direction of some of the stars closest to Earth. The distance and direction will not always be completely accurate, since for this game the stars are represented on two dimensions, as opposed to three dimensions of actual space. (There is assumed to be a good scientific reason for this.—Ed.)

Ship expenses must be paid every time a new starport is reached. If your account becomes negative after paying these expenses, you must sell enough cargo to make it positive before you can leave the starport. If you do not have enough cargo to do this, you are out of business and the game ends.

The expenses which must be paid consist of a docking fee, fuel expenses, and crew salary. The docking fee will always be 50,000 credits. The fuel expenses are directly proportional to the distance traveled from the last starport. The cost of fuel per lightyear is 100,000 credits. The crew's salary is based on an annual salary of 500,000 credits and the amount of ship time that has passed since the crew was last paid. Ship time increases approximately .02 years a week for each lightyear traveled and approximately .003 years a day for each cargo transaction.

As your fortune grows, the probability increases that the crew will go on strike for a higher salary. When a strike occurs the crew presents their salary demands and you are asked for a counter-offer. The probability that your counter-offer will be accepted depends upon the amount that you offer and the number of counter-offers that have been rejected previously. Once the crew has rejected ten counter-offers, they will accept

Lloyd Johnson, 11847 Silhouette Pl., Florissant, MO 63033.

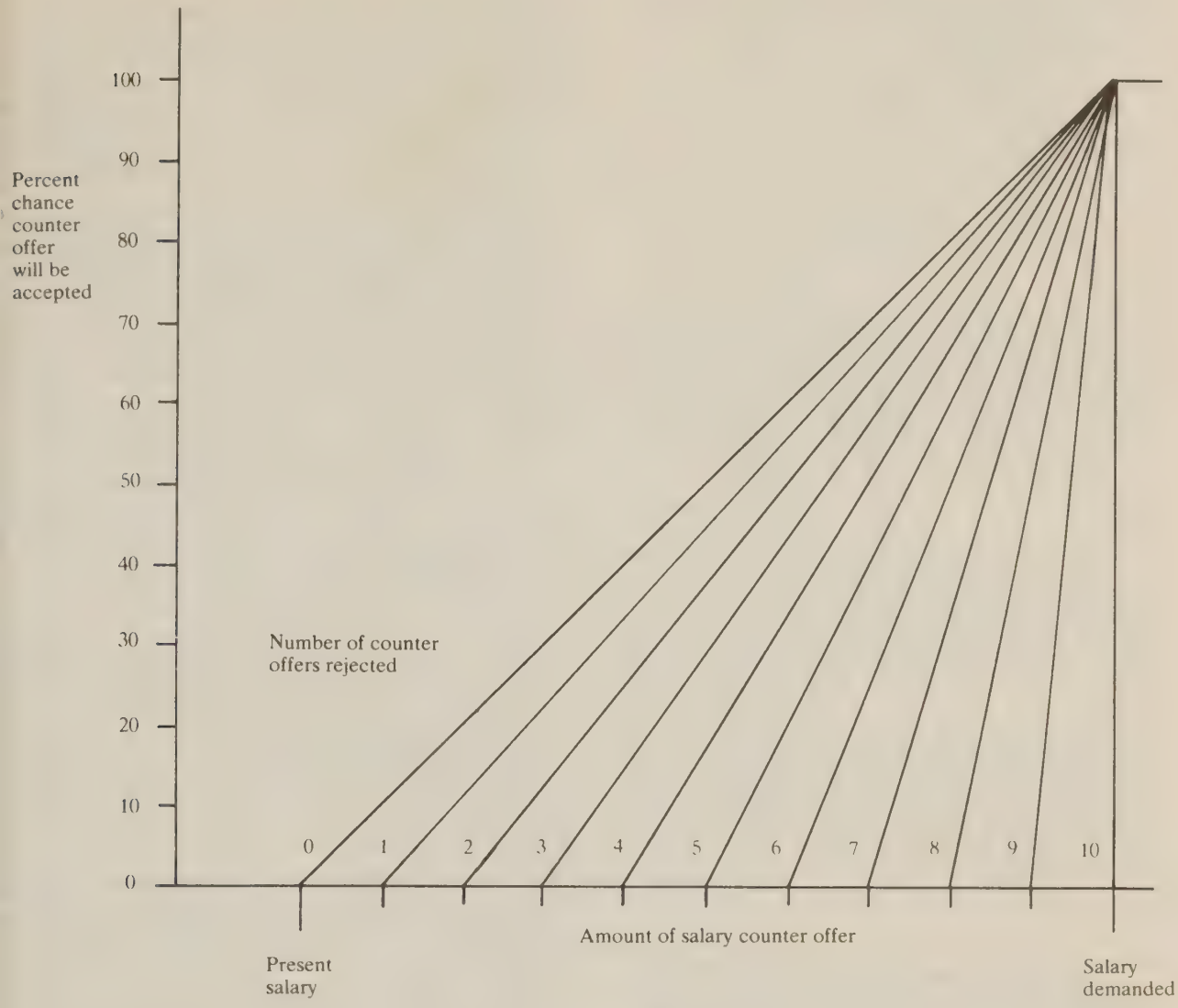


Figure 1.

only their original salary demands or higher. For this reason an early strike settlement is desirable.

The lease on your ship will expire after two years of ship time. At this time you will be asked to renew your lease if you have enough money. It will cost 2,000,000 credits for another two-year lease. Should you decide not to renew your lease, or if you do not have enough money for lease renewal, the game will end, and the game results will be displayed.

A copy of the program on cassette for the TRS-80 Level II computer may be obtained from me for $10.

Program Description

This program is written in H-P Basic. With little or no modification it should work on most machines. To aid in debugging, a detailed description of the program and the variables follows.

The program is arranged into five main sections. These sections and their corresponding subsections are listed as follows:

1. Initialize section. Lines 50-630.
2. Main logic section. Lines 640-880.
3. End program section. Lines 890-1030.
4. Subroutine section. Lines 1040-3640.
   - 4.1 Strike. Lines 1040-1230.
   - 4.2 List cargo in hold. Lines 1250-1410.
   - 4.3 List cargo for sale. Lines 1420-1580.
   - 4.4 Buy cargo. Lines 1590-1890.
   - 4.5 Sell cargo. Lines 1900-2040.
   - 4.6 List starports. Lines 2050-2140.
   - 4.7 Travel to new starport. Lines 2150-3370.
   - 4.8 List commands. Lines 3380-3460.

4.9 Restore READ statement to star data. Lines 3470-3520.
4.10 Restore READ statement to cargo data. Lines 3530-3580.
4.11 Restore READ statement to price data. Lines 3590-3640.

5. Data section. Lines 3650-4620.
5.1 Star names. Lines 3650-3750.
5.2 Cargo names. Lines 3770-4120.
5.3 Star data. Lines 4130-4230.
5.4 Cargo Data. Lines 4240-4600.
5.5 Price data. Lines 4610-4620.

The initialization section displays game instructions, initializes variables, seeds the random number function, and sets up the initial voyage from Alpha Centauri to Sol. (This initial voyage is necessary in the program logics since the statements that determine which cargos are for sale are contained in the travel to new starport subroutine.)

The main section is responsible for performing the following tasks:

1. Determining whether a strike is to occur.
2. Accepting a command and calling the appropriate subroutine.
3. Determining whether the player has run out of money.
4. Determining whether the ship's lease has expired.

Control is tranferred to the end section if the player runs out of money, if the lease on the ship expires and the player does not renew it, or if the lease expires and the player does not have enough money to renew it. If none of these conditions occur, control will be passed back to the beginning of the main section.

The end section prints out the end conditions of the game and transfers program control to the end statement.

The subroutine section contains subroutines which are called from both the initialization section and the main section. With the exception of subroutine 4.7 (travel to new starport), all subroutines are called by the line number listed at the beginning of the subroutine.

The ship time, player's account balance and available cargo space will be updated upon completion of the cargo transaction.

Upon entering the strike subroutine, the salary demands of the crew are calculated by adding from zero to fifty percent of their current salary to it. The minimum salary the crew will accept is then calculated from their current salary, their demanded salary, and the number of counter-offers which have previously been rejected. The minimum salary that the crew will accept will initially be equal to their current salary and will increase by ten percent of the difference between their salary demands and their current salary for each counter-offer that is rejected. The probability that a counter-offer will be accepted may be represented as a separate linear function for each number of counter-offers previously rejected. These functions are graphed in Figure 1.

The subroutines that list the cargos for sale and the cargos in the hold use the contents of arrays A and H respectively. A detailed description of these two arrays is included in the variable description. For each non-zero cargo type, the cargo name is obtained from the data section. This is done by doing a restore, followed by the required number of data reads necessary to position the data pointer to the cargo name that corresponds to the cargo. This cargo name along with other information in array A or H is then displayed with a PRINT USING statement. The PRINT USING statement and its corresponding IMAGE statement will require some modification in order for it to work with other Basics.

The subroutine for buying cargo consists of obtaining the lot number of the cargo to be purchased and transferring the corresponding data from the A array to the H array. The first empty position in the H array is used to store the cargo which has been purchased. The ship time, player's account balance, and available cargo space will be updated upon completion of the cargo transaction.

The subroutine for selling cargo consists of obtaining the partition number of the cargo to be sold, and sets the corresponding position in array H to zero. This is done to indicate that the partition is now empty. Ship time, player's account balance and available cargo will also be updated.

The list starport subroutine reads the starport name and trade class from the data section and displays them. Distance and direction for each starport will also be displayed, and stored in the arrays D and T respectively. The contents of these two arrays will have been previously calculated by the travel subroutine when it is executed.

The travel (to new starport) subroutine performs the following tasks:

1. Accept destination starport number and perform validity checks.
2. Get trade and location data for destination starport.
3. Appraise cargo stored in ship's hold.
4. Determine type, price and quantity of cargos available for purchase at destination starport.
5. Determine distance and direction of other starports from the destination starport.
6. Calculate and display arrival expenses and destination starport's name.

This subroutine is called each time the travel command is invoked, and also during program initialization. When this subroutine is called from the initialization section, the statements which accept the destination starport number are bypassed.

The three subroutines which are used to restore the READ statement to a particular data section basically do a restore followed by successive reads until the corresponding data section is reached.

The data section consists of the five types of data already mentioned. They are read as necessary. Each statement in the star-names section and the cargo-names section has a statement in the star-data section and the cargo-data section which corresponds to it.

The first two parameters in a data statement for star data are the X and Y coordinates of that star, relative to Sol. These values may be altered if you want to change the position of the star. The third parameter corresponds to the trade class of the starport. This value is calculated by representing each trade class by a power of two (listed in Table 1). An example of the calculation for generating the trade class is as follows: The trade class for Alpha Centauri is NA and I. The parameter representing this trade class is thus $2^2 + 2^3$, which equals 9.

The data statements for cargo data consist of eight parameters. The first six parameters are used in calculating the price of the cargo. If the trade class of the starport corresponds to the parameter, the value of the parameter will be added to the random number which determines the price. The trade class to which each parameter corresponds is listed in Table 1.

Trade Class	Power of Two	Parameter
Rich	5	1
Poor	4	2
Industrial	3	3
Nonindustrial	2	4
Agricultural	1	5
Nonagricultural	0	6

Table 1.

A(5,4) Available cargo for purchase. First subscript references one of the five available cargos.
A(I,1) represents the cargo type of available cargo I. Value is set to zero when cargo I is purchased.
A(I,2) represents the price of cargo I.
A(I,3) represents the weight of cargo I in tons.
A(I,4) represents the price of cargo I as a percentage of its base price.

B1 Base price of a cargo. Value for B1 is read from the data statements.

B2 Amount of money in player's account.

B3 Player's total assets. Equal to B2+T1.

D(10) Distance in lightyears from player's present location to each of the ten starports. D(5) is initialized to 4.3 for initial journey from Alpha Centauri to Sol.

D2 Price modifier calculated from cargo trade modifiers and starport trade class.

D3 Cargo trade modifiers read from data statements.

D4 Distance to the destination starport.

D9 Dummy variable. Value is never used.

E1 Docking expense. Initial value is 5E4.

E2 Fuel expense for light year traveled. Value is 1E5.

E3 Annual salary of crew. Initial value is 5E5.

E4 Amount of crew's salary paid upon arriving at a starport.

E5 Salary demands of crew.

E6 Minimum counter-offer crew will accept during a salary strike.

E7 Amount of salary counter-offer.

G1 Average annual profit gain calculated at the end of the game.

H(20,4) Cargo stored in ship's hold. First subscript references the cargo hold partition number.
H(I,1) represents the cargo type in partition I. Value is set to zero when partition is empty.
H(I,2) represents the price of the cargo in partition I.
H(I,3) represents the weight of the cargo in partition I.
H(I,4) represents the percentage of the base price the cargo in partition I has been appraised at.

I Temporary variable.

J Temporary variable.

K Temporary variable.

L Temporary variable.

M1 Trade class parameter of Starport.

M2 Variable used in unpacking trade class parameter, M1.

N1 Number of cargos available for purchase.

N2 Number of cargo types. Initialized to 36.

N3 Number of starports. Initialized to 10.

N4 Number of counter-offers that have been rejected.

P(14) Percentage values read from data statement. Used in modifying the base price of a cargo.

P2 Percentage index. Values may range from 1 to 14. Used to index P(14) to determine the percent of base price at which the cargo will trade.

Q1 Quantity index of a cargo type, as read from cargo data. Represents the number of times a random between 1 and 6 is used to generate the quantity.

Q2 Quantity multiplier. Random number used in generating cargo quantity is multiplied by Q2.
Q2 = 1 for cargo types 1 through 16.
Q2 = 5 for cargo types 17 through 31.
Q2 = 10 for cargo types 32 through 36.

Q3 Quantity of cargo in tons.

R1 Argument for random number function. Set equal to zero.

S(6) Array containing the trade class of the destination starport. Value is either 1 or 0.
S(1) = 1 implies starport is rich.
S(2) = 1 implies starport is poor.
S(3) = 1 implies starport is industrial.
S(4) = 1 implies starport is nonindustrial.
S(5) = 1 implies starport is agricultural.
S(6) = 1 implies starport is nonagricultural.

S1 Destination starport for starship. Initialized to 5.

T(10) Direction in degrees of all ten starports.

T1 Total value of cargo stored in the hold.

T2 Cargo type.

T5 The number of times the ship's two year lease is renewed.

T6 Elapsed ship time in years.

T7 Ship time when crew was paid.

W1 Total cargo capacity of ship. Set to 200.

W2 Amount of empty cargo space on ship. Initially set equal to W1 to indicate an empty cargo hold.

X1 Starport's X coordinate relative to Sol.

X2 Starport's X coordinate relative to present position of ship.

X3 X coordinate of destination star relative to Sol.

Y1 Y coordinate of starport relative to Sol.

Y2 Y coordinate of starport relative to ship's present position.

Y3 Y coordinate of destination star relative to Sol.

Star Merchant Sample Run

```
                    STAR MERCHANT
YOU HAVE JUST SPENT 2 MILLION CREDITS ON A 2 YEAR
LEASE FOR A MERCHANT STARSHIP.  THIS LEAVES YOU WITH
2 MILLION CREDITS OPERATING CAPITAL.

YOUR SHIP CAN HOLD A TOTAL OF 20 CARGOS WITH A TOTAL
CARGO WEIGHT OF 200 TONS.  THE FUEL CAPACITY OF YOUR
SHIP IS GREAT ENOUGH SUCH THAT TRAVEL BETWEEN ANY
TWO STARPORTS IS POSSIBLE WITHOUT REFUELING.

YOU ARE PRESENTLY TRAVELING FROM ALPHA CENTAURI TO SOL.
YOU ARE CARRYING NO CARGO.

THE STARPORT TRADE CLASSIFICATION DETERMINES THE CARGO
PRICE BUT DOES NOT DETERMINE WHICH CARGOS ARE AVAILABLE.

ABBREVIATIONS USED FOR TRADE CLASS ARE AS FOLLOWS:
R-RICH,P-POOR,I-INDUSTRIAL,NI-NONINDUSTRIAL,
A-AGRICULTURAL,NA-NONAGRICULTURAL

PRESS RETURN TO CONTINUE?109
COMMANDS AVAILABLE ARE AS FOLLOWS:
     COMMAND              DESCRIPTION
        1                 LIST CARGO IN HOLD
        2                 LIST CARGO WHICH MAY BE PURCHASED
        3                 BUY CARGO
        4                 SELL CARGO
        5                 LIST STARPORTS
        6                 TRAVEL TO NEW STAR
      OTHER               LIST AVAILABLE COMMANDS

INPUT A RANDOM NUMBER BETWEEN 1 AND 500?109


YOU HAVE ARRIVED AT SOL
EXPENSES HAVE BEEN DEDUCTED AS FOLLOWS:
DOCKING FEE 50000.
FUEL 430000.
CREW SALARY 53000.

ACCOUNT BALANCE 1.46700E+06    EMPTY CARGO SPACE 200
SHIP TIME .106         YEARS.    ENTER COMMAND?2

CARGOS AVAILABLE FOR PURCHASE
CARGO NUM      DESCRIPTION      AMT     PRICE    % OF BASE
    1      LIQUOR                25      375000      150
    2      RADIOACTIVES           6     5400000       90
    3      GRAV SLEDS             4    31200000      130
    4      ELECTRONIC PARTS       5      750000      150
    5      POLYMERS              40      196000       70

ACCOUNT BALANCE 1.46700E+06    EMPTY CARGO SPACE 200
SHIP TIME .106         YEARS.    ENTER COMMAND?3

ENTER THE LOT NUMBER OF CARGO YOU WISH TO PURCHASE?5
TRANSACTION COMPLETED
CARGO STORED IN PARTITION 1

ACCOUNT BALANCE 1.27100E+06    EMPTY CARGO SPACE 160
SHIP TIME .109         YEARS.    ENTER COMMAND?5

STAR NUM         NAME          TRADE CLASS  DISTANCE DIRECTION
    1     LALANDE 21185          NI,P          7.89        291
    2     ALPHA CENTAURI         NA,I          4.29        236
    3     SIRUS                  A            12.99         50
    4     BARNARDS STAR          I,P           6.10        180
    5     SOL                    R             0.00         90
    6     ROSS 154               NI,NA         9.10        167
    7     EPSILON ERIDANI        A,P          10.75         38
    8     LUYTEN 726-8           NA            8.00         68
    9     LUYTEN 789-6           A,NI,P       10.30        115
   10     ROSS 248               A,I          10.30         95

ACCOUNT BALANCE 1.27100E+06    EMPTY CARGO SPACE 160
SHIP TIME .109         YEARS.    ENTER COMMAND?6

ENTER DESTINATION STAR NUMBER?2

YOU HAVE ARRIVED AT ALPHA CENTAURI
EXPENSES HAVE BEEN DEDUCTED AS FOLLOWS:
DOCKING FEE 50000.
FUEL 429344.
CREW SALARY 54434.4

ACCOUNT BALANCE 737222.      EMPTY CARGO SPACE 160
SHIP TIME .214869      YEARS.    ENTER COMMAND?1
```

```
CARGO STORED IN HOLD
PARTITION      DESCRIPTION      AMT     PRICE    % OF BASE
    1          POLYMERS         40      476000     170

ACCOUNT BALANCE 737222.      EMPTY CARGO SPACE 160
SHIP TIME .214869      YEARS.     ENTER COMMAND?4

ENTER PARTITION OF CARGO TO BE SOLD?1
TRANSACTION COMPLETED

THE CREW HAS GONE ON STRIKE
YOU ARE CURRENTLY PAYING THEM 500000.      CREDITS ANNUALLY
THEY ARE ASKING FOR 747858.      CREDITS ANNUAL SALARY

ENTER COUNTER OFFER?660000.
OFFER ACCEPTED

ACCOUNT BALANCE 1.21322E+06     EMPTY CARGO SPACE 200
SHIP TIME .217869      YEARS.     ENTER COMMAND?2

CARGOS AVAILABLE FOR PURCHASE
CARGO NUM      DESCRIPTION      AMT     PRICE    % OF BASE
    1          GRAIN           100      51000     170
    2          COMPUTER PARTS   20    2100000      70
    3          GRAIN           165     198000     400
    4          GRAV SLEDS        5   32999995     110
    5          AIRCRAFT          5    2000000      40
ACCOUNT BALANCE 1.21322E+06     EMPTY CARGO SPACE 200
SHIP TIME .217869      YEARS.     ENTER COMMAND?3

ENTER THE LOT NUMBER OF CARGO YOU WISH TO PURCHASE?1
TRANSACTION COMPLETED
CARGO STORED IN PARTITION 1

ACCOUNT BALANCE 1.16222E+06     EMPTY CARGO SPACE 100
SHIP TIME .220869      YEARS.     ENTER COMMAND?5

STAR NUM          NAME          TRADE CLASS  DISTANCE DIRECTION
    1      LALANDE 21185          NI,P          6.45          324
    2      ALPHA CENTAURI         NA,I          0.00           90
    3      SIRUS                  A            17.27           51
    4      BARNARDS STAR          I,P           5.13          136
    5      SOL                    R             4.29           56
    6      ROSS 154               NI,NA         8.56          139
    7      EPSILON ERIDANI        A,P          10.90           43
    8      LUYTEN 726-8           NA           12.23           64
    9      LUYTEN 789-6           A,NI,P       13.02           99
   10      ROSS 248               A,I          13.90           84
ACCOUNT BALANCE 1.16222E+06     EMPTY CARGO SPACE 100
SHIP TIME .220869      YEARS.     ENTER COMMAND?6

ENTER DESTINATION STAR NUMBER?4

YOU HAVE ARRIVED AT BARNARDS STAR
EXPENSES HAVE BEEN DEDUCTED AS FOLLOWS:
DOCKING FEE 50000.
FUEL 513455.
CREW SALARY 84936.

ACCOUNT BALANCE 513831.      EMPTY CARGO SPACE 100
SHIP TIME .34356       YEARS.     ENTER COMMAND?1

CARGO STORED IN HOLD
PARTITION      DESCRIPTION      AMT     PRICE    % OF BASE
    1          GRAIN           100      36000     120

ACCOUNT BALANCE 513831.      EMPTY CARGO SPACE 100
SHIP TIME .34356       YEARS.     ENTER COMMAND?2

CARGOS AVAILABLE FOR PURCHASE
CARGO NUM      DESCRIPTION      AMT     PRICE    % OF BASE
    1          BODY ARMOR        6     300000     100
    2          CRYSTALS          1      10000      50
    3          MECHANICAL PARTS 20    1050000      70
    4          GRAV SLEDS        3   14400000      80
    5          CYBERNETIC PARTS 15    1500000      40

ACCOUNT BALANCE 513831.      EMPTY CARGO SPACE 100
SHIP TIME .34356       YEARS.     ENTER COMMAND?3

ENTER THE LOT NUMBER OF CARGO YOU WISH TO PURCHASE?2
TRANSACTION COMPLETED
CARGO STORED IN PARTITION 2
ACCOUNT BALANCE 503831.      EMPTY CARGO SPACE 99
SHIP TIME .34656       YEARS.     ENTER COMMAND?3
ENTER THE LOT NUMBER OF CARGO YOU WISH TO PURCHASE?1
TRANSACTION COMPLETED
CARGO STORED IN PARTITION 3
```

Star Merchant Program Listing

```
10  REM
20  REM STAR MERCHANT
30  REM BY LLOYD JOHNSON
40  REM
50  REM INITIALIZE SECTION
60  DIM A$[20],B$[8],A[5,4],D[10],H[20,4]
70  DIM P[14],S[6],T[10]
80  PRINT TAB(20)"STAR MERCHANT"
90  PRINT "YOU HAVE JUST SPENT 2 MILLION CREDITS ON A 2 YEAR"
100  PRINT "LEASE FOR A MERCHANT STARSHIP.  THIS LEAVES YOU WITH"
110  PRINT "2 MILLION CREDITS OPERATING CAPITAL."
120  PRINT
130  PRINT "YOUR SHIP CAN HOLD A TOTAL OF 20 CARGOS WITH A TOTAL"
140  PRINT "CARGO WEIGHT OF 200 TONS.  THE FUEL CAPACITY OF YOUR"
150  PRINT "SHIP IS GREAT ENOUGH SUCH THAT TRAVEL BETWEEN ANY"
160  PRINT "TWO STARPORTS IS POSSIBLE WITHOUT REFUELING."
170  PRINT
180  PRINT "YOU ARE PRESENTLY TRAVELING FROM ALPHA CENTAURI TO SOL."
190  PRINT "YOU ARE CARRYING NO CARGO."
200  PRINT
210  PRINT "THE STARPORT TRADE CLASSIFICATION DETERMINES THE CARGO"
220  PRINT "PRICE BUT DOES NOT DETERMINE WHICH CARGOS ARE AVAILABLE."
230  PRINT
240  PRINT "ABBREVIATIONS USED FOR TRADE CLASS ARE AS FOLLOWS:"
250  PRINT "R=RICH,P=POOR,I=INDUSTRIAL,NI=NONINDUSTRIAL,"
260  PRINT "A=AGRICULTURAL,NA=NONAGRICULTURAL"
270  PRINT
280  PRINT "PRESS RETURN TO CONTINUE";
290  INPUT A$
300  PRINT "COMMANDS AVAILABLE ARE AS FOLLOWS:"
310  GOSUB 3380
320  LET R1=0
330  PRINT
340  PRINT "INPUT A RANDOM NUMBER BETWEEN 1 AND 500";
350  INPUT I
360  PRINT
370    FOR J=1 TO I
380    LET D9=RND(R1)
390    NEXT J
400  DEF FNA(R1)=INT(6*RND(R1))+1
410  DEF FNB(N2)=INT(N2*RND(R1))+1
420  DEF FNC(R1)=FNA(R1)+FNA(R1)
430  LET B2=2.00000E+06
440  LET D[5]=4.3
450  LET S1=5
460  LET E1=50000.
470  LET E2=100000.
480  LET E3=500000.
490  LET N2=36
500  LET N3=10
510  LET W1=200
520  LET W2=W1
530    FOR I=1 TO 20
540    LET H[I,1]=0
550    NEXT I
560  GOSUB 3590
570    FOR I=1 TO 14
580    READ P[I]
590    NEXT I
600  LET T5=0
610  LET T6=0
620  LET T7=0
630  GOSUB 2280
640  REM MAIN LOGIC
650  IF RND(R1)>.01+.005*B2/E3 THEN 670
660  GOSUB 1040
670  PRINT
680  PRINT "ACCOUNT BALANCE ";B2;"EMPTY CARGO SPACE ";W2
690  PRINT "SHIP TIME ";T6;"YEARS.    ENTER COMMAND";
700  INPUT C1
710  PRINT
720  IF C1>0 AND C1<7 AND C1=INT(ABS(C1)) THEN 750
730  GOSUB 3380
740  GOTO 640
750  GOSUB C1 OF 1240,1420,1590,1900,2050,2150
760  LET B3=B2+T1
770  IF B3<0 THEN 890
780  IF T6=2*T5<2 THEN 640
790  PRINT
800  PRINT "THE LEASE HAS EXPIRED ON YOUR SHIP"
```

```
810  IF B3<2.00000E+06 THEN 890
820  PRINT "ANOTHER TWO YEAR LEASE WILL COST 2E6 CREDITS"
830  PRINT "DO YOU WISH TO RENEW YOUR LEASE(Y/N)";
840  INPUT A$
850  IF A$="N" THEN 890
860  LET B2=B2-2.00000E+06
870  LET T5=T5+1
880  GOTO 640
890  REM END PROGRAM
900  LET G1=(B3-4.00000E+06)/T6
910  LET A$="LOSS"
920  IF G1<0 THEN 940
930  LET A$="GAIN"
940  IF B3>0 THEN 970
950  PRINT
960  PRINT "YOU NO LONGER HAVE SUFFICIENT FUNDS TO OPERATE YOUR SHIP"
970  PRINT
980  PRINT "YOU STARTED WITH 4E6 CREDITS"
990  PRINT "YOU NOW HAVE ";B2
1000  PRINT "CARGO IN HOLD IS WORTH ";T1
1010  PRINT
1020  PRINT "THIS REPRESENTS A ";A$;" OF ";G1;"CREDITS PER YEAR"
1030  GOTO 4630
1040  REM STRIKE SUB
1050  PRINT
1060  PRINT "THE CREW HAS GONE ON STRIKE"
1070  PRINT "YOU ARE CURRENTLY PAYING THEM ";E3;"CREDITS ANNUALLY"
1080  LET E5=E3+.5*RND(R1)*E3
1090  PRINT "THEY ARE ASKING FOR ";E5;"CREDITS ANNUAL SALARY"
1100  LET N4=0
1110  LET E6=E3+N4*(E5-E3)/10
1120  PRINT
1130  PRINT "ENTER COUNTER OFFER";
1140  INPUT E7
1150  IF E7 >= E5 THEN 1210
1160  IF E7<E6 THEN 1180
1170  IF (E5-E7)/(E5-E6)<RND(R1) THEN 1210
1180  PRINT "OFFER REJECTED - TRY AGAIN"
1190  LET N4=N4+1
1200  GOTO 1110
1210  PRINT "OFFER ACCEPTED"
1220  LET E3=E7
1230  RETURN
1240  REM LIST HOLD
1250  PRINT "CARGO STORED IN HOLD"
1260  IF N2<W1 THEN 1290
1270  PRINT "ALL PARTITIONS ARE EMPTY"
1280  RETURN
1290  PRINT "PARTITION      DESCRIPTION      AMT";
1300  PRINT "    PRICE   % OF BASE"
1310    FOR I=1 TO 20
1320    RESTORE
1330    IF H[I,1]=0 THEN 1400
1340      FOR J=1 TO 19+H[I,1]
1350      READ A$
1360      NEXT J
1370    READ A$
1380    PRINT  USING 1390;I,A$,H[I,3],H[I,2],H[I,4]
1390    IMAGE 3X,2D,5X,18A,2X,3D,2X,9D,4X,3D
1400    NEXT I
1410  RETURN
1420  REM LIST CARGO FOR SALE
1430  PRINT "CARGOS AVAILABLE FOR PURCHASE"
1440  IF N1>0 THEN 1470
1450  PRINT "ALL AVAILABLE CARGOS HAVE BEEN PURCHASED"
1460  RETURN
1470  PRINT "CARGO NUM      DESCRIPTION      AMT";
1480  PRINT "    PRICE   % OF BASE"
1490    FOR I=1 TO 5
1500    RESTORE
1510    IF A[I,1]=0 THEN 1570
1520      FOR J=1 TO 19+A[I,1]
1530      READ A$
1540      NEXT J
1550    READ A$
1560    PRINT  USING 1390;I,A$,A[I,3],A[I,2],A[I,4]
1570    NEXT I
1580  RETURN
1590  REM BUY CARGO SUB
1600  PRINT "ENTER THE LOT NUMBER OF CARGO YOU WISH TO PURCHASE";
1610  INPUT K
1620  IF K=ABS(INT(K)) AND K>0 AND K<6 THEN 1650
1630  PRINT "INVALID LOT NUMBER"
1640  RETURN
1650  IF A[K,1]>0 THEN 1680
1660  PRINT "LOT ";K;"HAS BEEN ALREADY PURCHASED"
1670  RETURN
1680  IF A[K,2] <= B2 THEN 1710
```

```
1690  PRINT "YOU CANNOT BUY CARGO ON CREDIT"
1700  RETURN
1710  IF A[K,3] <= W2 THEN 1740
1720  PRINT "YOU DO NOT HAVE SUFFICIENT CARGO SPACE"
1730  RETURN
1740    FOR I=1 TO 20
1750    IF H[I,1]=0 THEN 1790
1760    NEXT I
1770  PRINT "ALL 20 CARGO PARTITIONS ARE OCCUPIED"
1780  RETURN
1790    FOR J=1 TO 4
1800    LET H[I,J]=A[K,J]
1810    NEXT J
1820  LET A[K,1]=0
1830  LET W2=W2-H[I,3]
1840  LET B2=B2-H[I,2]
1850  LET T6=T6+.003
1860  LET N1=N1-1
1870  PRINT "TRANSACTION COMPLETED"
1880  PRINT "CARGO STORED IN PARTITION ";I
1890  RETURN
1900  REM SELL CARGO SUB
1910  PRINT "ENTER PARTITION OF CARGO TO BE SOLD";
1920  INPUT K
1930  IF K=ABS(INT(K)) AND K>0 AND K<21 THEN 1960
1940  PRINT "INVALID PARTITION NUMBER"
1950  RETURN
1960  IF H[K,1]>0 THEN 1990
1970  PRINT "CARGO PARTITION IS EMPTY"
1980  RETURN
1990  LET B2=B2+H[K,2]
2000  LET W2=W2+H[K,3]
2010  LET T6=T6+.003
2020  LET H[K,1]=0
2030  PRINT "TRANSACTION COMPLETED"
2040  RETURN
2050  REM LIST STARPORTS SUB
2060  PRINT "STAR NUM          NAME          TRADE CLASS";
2070  PRINT "  DISTANCE DIRECTION"
2080  RESTORE
2090    FOR I=1 TO 10
2100    READ A$,B$
2110    PRINT  USING 2120;I,A$,B$,D[I],T[I]
2120    IMAGE 3X,2D,4X,20A,2X,10A,3D.2D,9X,3D
2130    NEXT I
2140  RETURN
2150  REM TRAVEL SUB
2160  IF B2>0 THEN 2190
2170  PRINT "YOU CANNOT LEAVE STARPORT
UNTIL ALL DEBTS ARE CLEARED"
2180  RETURN
2190  PRINT "ENTER DESTINATION STAR NUMBER";
2200  INPUT I
2210  IF I#S1 THEN 2240
2220  PRINT "YOU ARE ALREADY AT ";I
2230  RETURN
2240  IF I=ABS(INT(I)) AND I>0 AND I<N3+1 THEN 2270
2250  PRINT "INVALID STAR NUMBER"
2260  RETURN
2270  LET S1=I
2280  REM GET STAR TRADE & LOCATION DATA
2290  GOSUB 3470
2300  IF S1=1 THEN 2340
2310    FOR I=1 TO (S1-1)*3
2320    READ D9
2330    NEXT I
2340  READ X3,Y3,M1
2350    FOR I=1 TO 6
2360    LET M2=2^(6-I)
2370    LET S[I]=INT(M1/M2)
2380    LET M1=M1-S[I]*M2
2390    NEXT I
2400  REM APPRAISE CARGO IN HOLD
2410  LET T1=0
2420    FOR I=1 TO 20
2430    IF H[I,1]=0 THEN 2630
2440    GOSUB 3530
2450    IF H[I,1]=1 THEN 2490
2460      FOR J=1 TO (H[I,1]-1)*8
2470      READ D9
2480      NEXT J
2490    LET D2=0
2500      FOR J=1 TO 6
2510      READ D3
2520      LET D2=D2+D3*S[J]
2530      NEXT J
2540    READ B1
2550    LET P2=FNC(R1)+D2-1
2560    IF P2>1 THEN 2580
2570      LET P2=1
2580      IF P2<14 THEN 2600
2590      LET P2=14
2600      LET H[I,4]=P[P2]*100
2610      LET H[I,2]=P[P2]*B1*H[I,3]
2620      LET T1=T1+H[I,2]
2630      NEXT I
2640  REM GET CARGOS FOR SALE
2650  LET N1=5
2660    FOR I=1 TO 5
2670    LET D2=0
2680    LET T2=FNB(N2)
2690    LET A[I,1]=T2
2700    GOSUB 3530
2710    IF T2=1 THEN 2750
2720      FOR J=1 TO (T2-1)*8
2730      READ D9
2740      NEXT J
2750      FOR J=1 TO 6
2760      READ D3
2770      LET D2=D2+D3*S[J]
2780      NEXT J
2790    LET P2=FNC(R1)+D2-1
2800    IF P2>1 THEN 2820
2810    LET P2=1
2820    IF P2<14 THEN 2840
2830    LET P2=14
2840    LET A[I,4]=P[P2]*100
2850    READ B1,Q1
2860    LET Q2=1
2870    IF T2<17 THEN 2910
2880    LET Q2=5
2890    IF T2<32 THEN 2910
2900    LET Q2=10
2910    LET Q3=0
2920      FOR J=1 TO Q1
2930      LET Q3=FNA(R1)*Q2+Q3
2940      NEXT J
2950    LET A[I,3]=Q3
2960    LET A[I,2]=B1*Q3*P[P2]
2970    NEXT I
2980  REM DISTANCE AND DIRECTION OF STARS
2990  LET D4=D[S1]
3000  GOSUB 3470
3010    FOR I=1 TO 10
3020    READ X1,Y1,D9
3030    LET X2=X1-X3
3040    LET Y2=Y1-Y3
3050    IF X2#0 THEN 3110
3060    IF Y2<0 THEN 3090
3070    LET T[I]=90
3080    GOTO 3180
3090    LET T[I]=270
3100    GOTO 3180
3110    LET T[I]=ATN(Y2/X2)*180/3.14159
3120    IF X2>0 THEN 3140
3130    LET T[I]=T[I]+180
3140    IF T[I]<360 THEN 3160
3150    LET T[I]=T[I]-360
3160    IF T[I]>0 THEN 3180
3170    LET T[I]=T[I]+360
3180    LET D[I]=SQR(X2^2+Y2^2)
3190    NEXT I
3200  REM ARRIVAL EXPENSES AND STAR NAME
3210  LET T6=T6+.02*D4+.02
3220  LET E4=(T6-T7)*E3
3230  LET T7=T6
3240  RESTORE
3250  IF S1=1 THEN 3290
3260    FOR I=1 TO S1-1
3270    READ A$,B$
3280    NEXT I
3290  READ A$,B$
3300  PRINT
3310  PRINT "YOU HAVE ARRIVED AT ";A$
3320  PRINT "EXPENSES HAVE BEEN DEDUCTED AS FOLLOWS:"
3330  PRINT "DOCKING FEE ";E1
3340  PRINT "FUEL ";E2*D4
3350  PRINT "CREW SALARY ";E4
3360  LET B2=B2-(E1+E2*D4+E4)
3370  RETURN
3380  PRINT TAB(5);"COMMAND";TAB(25);"DESCRIPTION"
3390  PRINT TAB(8);"1";TAB(25);"LIST CARGO IN HOLD"
3400  PRINT TAB(8);"2";TAB(25);"LIST CARGO WHICH
MAY BE PURCHASED"
3410  PRINT TAB(8);"3";TAB(25);"BUY CARGO"
3420  PRINT TAB(8);"4";TAB(25);"SELL CARGO"
3430  PRINT TAB(8);"5";TAB(25);"LIST STARPORTS"
3440  PRINT TAB(8);"6";TAB(25);"TRAVEL TO NEW STAR"
```

```
3610    FOR L=1 TO 8*N2
3620    READ D9
3630    NEXT L
3640  RETURN
3650  REM STAR NAMES
3660  DATA "LALANDE 21185","NI,P"
3670  DATA "ALPHA CENTAURI","NA,I"
3680  DATA "SIRUS","A"
3690  DATA "BARNARDS STAR","I,P"
3700  DATA "SOL","R"
3710  DATA "ROSS 154","NI,NA"
3720  DATA "EPSILON ERIDANI","A,P"
3730  DATA "LUYTEN 726-8","NA"
3740  DATA "LUYTEN 789-6","A,NI,P"
3750  DATA "ROSS 248","A,I"
3760  REM CARGO NAMES
3770  DATA "CRYSTALS"
3780  DATA "RADIOACTIVES"
3790  DATA "SPECIAL ALLOYS"
3800  DATA "PHARMACEUTICALS"
3810  DATA "GEMS"
3820  DATA "AIRCRAFT"
3830  DATA "GRAV SLEDS"
3840  DATA "COMPUTERS"
3850  DATA "ATV"
3860  DATA "AFV"
3870  DATA "FIREARMS"
3880  DATA "AMMUNITION"
3450  PRINT TAB(6);"OTHER";TAB(25);"LIST AVAILABLE COMMANDS"
3460  RETURN
3470  REM RESTORE TO STAR DATA SUB
3480  RESTORE
3490    FOR L=1 TO (2*N3+N2)
3500    READ A$
3510    NEXT L
3520  RETURN
3530  REM RESTORE TO CARGO DATA SUB
3540  GOSUB 3470
3550    FOR L=1 TO 3*N3
3560    READ D9
3570    NEXT L
3580  RETURN
3590  REM RESTORE TO PRICE DATA SUB
3600  GOSUB 3530
3890  DATA "PLASMA GUNS"
3900  DATA "TOOLS"
3910  DATA "BODY ARMOR"
3920  DATA "FARM MACHINERY"
3930  DATA "LIQUOR"
3940  DATA "SILVER"
3950  DATA "SPICES"
3960  DATA "ELECTRONIC PARTS"
3970  DATA "MECHANICAL PARTS"
3980  DATA "CYBERNETIC PARTS"
3990  DATA "COMPUTER PARTS"
4000  DATA "MACHINE TOOLS"
4010  DATA "SPACE SUITS"
4020  DATA "FRUIT"
4030  DATA "TEXTILES"
4040  DATA "POLYMERS"
4050  DATA "MEAT"
4060  DATA "PETROCHEMICALS"
4070  DATA "GRAIN"
4080  DATA "WOOD"
4090  DATA "COPPER"
4100  DATA "TIN"
4110  DATA "STEEL"
4120  DATA "ALUMINUM"
4130  REM STAR DATA
4140  DATA 2.83,-7.36,20
4150  DATA -2.4,-3.56,9
4160  DATA 8.38,9.93,2
4170  DATA -6.1,0,24
4180  DATA 0,0,32
4190  DATA -8.87,2.05,5
4200  DATA 8.45,6.65,18
4210  DATA 2.99,7.42,1
4220  DATA -4.43,9.3,22
4230  DATA -.89,10.26,10
4240  REM CARGO DATA
4250  DATA 3,-2,2,-2,0,-4,20000,1
4260  DATA 0,1,4,-3,0,-2,1.00000E+06,1
4270  DATA -2,0,-4,6,1,-2,200000.,1
4280  DATA -1,4,-4,3,-2,0,100000.,1
4290  DATA 4,-2,4,-4,-1,1,1.00000E+06,1
4300  DATA -2,4,-3,3,1,-1,1.00000E+06,1
4310  DATA 2,0,-1,1,0,0,6.00000E+06,1
4320  DATA 1,0,-2,0,0,0,1.00000E+07,1
4330  DATA -2,2,-2,1,1,0,300000.,1
4340  DATA 0,2,-2,0,0,1,700000.,1
4350  DATA -2,6,-4,1,0,0,30000,2
4360  DATA -1,6,-5,2,0,0,30000,2
4370  DATA -1,3,-2,0,0,0,200000.,2
4380  DATA -4,7,-8,4,5,0,10000,2
4390  DATA -3,6,-4,1,0,0,50000.,2
4400  DATA -2,2,-6,0,6,-4,150000.,1
4410  DATA 3,3,-1,0,-3,0,10000,1
4420  DATA 3,-1,3,-1,0,-2,70000.,1
4430  DATA 4,-2,3,-1,-5,2,6000,1
4440  DATA 0,0,-4,4,1,1,100000.,1
4450  DATA 0,1,-3,3,2,1,75000.,1
4460  DATA 1,0,-4,2,1,0,250000.,1
4470  DATA -1,0,-2,3,0,0,150000.,1
4480  DATA 1,0,-2,1,0,0,750000.,1
4490  DATA -1,2,-3,2,2,0,400000.,1
4500  DATA 1,2,3,3,-4,-6,1000,2
4510  DATA 3,0,-3,1,-5,-3,3000,3
4520  DATA -2,0,3,3,0,0,7000,4
4530  DATA 0,0,5,2,-5,5,1500,4
4540  DATA 2,0,4,-2,3,0,10000,6
4550  DATA 0,0,1,3,-5,6,300,8
4560  DATA 0,0,1,2,-7,3,1000,2
4570  DATA 2,2,3,-2,-1,-4,2000,2
4580  DATA 2,2,3,-4,-1,-2,9000,3
4590  DATA -1,2,6,0,0,0,500,4
4600  DATA -1,1,3,-2,0,-2,1000,5
4610  REM PRICE DATA
4620  DATA .4,.5,.7,.8,.9,1,1.1,
1.2,1.3,1.5,1.7,2,3,4
4630  END
```

Widget

Gary McGath

Is it possible for a game to present an interesting challenge, yet fit in 1K of RAM? As evidence that the answer is "yes," I offer *Widget*, a relative of the games *Hammurabi* and *Lemonade Stand*, in which the player allocates his resources for the best effect. As the player, you are the head of the United Widget Company, trying to expand your business as rapidly as possible without going broke.

You start out with one plant and $90,000. A plant is capable of producing as many as 1000 widgets a month, which will sell for $10 each. The monthly fixed overhead is $4000, including the mortgage cost for your first plant. In addition, it costs $5 per widget to produce any widgets in excess of the first hundred. To sell any widgets, you have to advertise. Naturally, the return on advertising isn't entirely predictable, but you have to learn how much to spend in order to sell what you produce at a profit.

The key to making money is buying more plants. A plant costs $100,000 to buy and $2000 a month to maintain, and it lets you produce another thousand widgets a month. This means more fixed cost and more need for advertising; it also means economies of scale. You'll notice that as production increases, it takes more advertising dollars to sell each widget. On the other hand, if you keep a steady flow coming out of your plants, each month's sales will produce a carry-over effect into the next month.

Gary McGath, 5 Ames Rd., RFD #3, Milford, NH 03055.

With these costs in mind, you can start playing *Widget*. Each turn represents one month. At the start of the turn. the screen shows your current cash balance, number of plants, and inventory. If you have at least $100,000, you are asked, "Buy a plant?" To buy a plant, answer "Y" (just the one letter); anything else means "no."

Next (or first, if you had less than $100,000), the program asks for your ad budget. Enter the number of *thousands* of dollars that you want to spend on ads; for example, to spend $4000, enter "4." You will be asked again for input if you try to spend more than you have.

Finally you are asked how many widgets you want to produce. Enter the number of *hundreds* of widgets you want to produce. remembering that each plant can only produce 1000 widgets. Again, the program won't let you go until you've entered a legal number.

The advertising cost is then subtracted from your cash balance, sales are determined, revenue comes in. and then the other costs are subtracted. If you can't meet your costs, the message "You are bankrupt" appears, and the game is over. Otherwise, the program goes on to the next month.

Before starting, decide how many months you're going to play. Sixty turns (5 years) or 120 turns (10 years) make a good game. Keep score by calculating your company's net worth, which is its cash balance plus $100,000 per plant. There isn't any predetermined winning score, but the outcome of a fairly well-played game would be to have 5 plants and $50,000 or more at the end of 10 years.

Sinclair ZX80

```
 5 RANDOMIZE
 6 LET Z = 1
10 LET P = 1
20 LET M = 990
30 LET I = 0
40 LET S = 10
100 CLS
110 PRINT "MONTH" ,Z
120 LET Z = Z + 1
300 PRINT "YOU HAVE" , "$" ;M; "00"
310 PRINT ,P;" PLANTS"
320 PRINT ,I;"00 WIDGETS"
330 PRINT "LAST MONTH YOU
          SOLD ";S; "00"
350 IF M < 1000 THEN GO TO 410
360 PRINT "BUY A PLANT?"
370 INPUT Y$
380 IF NOT Y$ = "Y" THEN GO TO 410
390 LET P = P + 1
400 LET M = M - 1000
410 PRINT "AD BUDGET IN 1000S ?"
420 INPUT B
430 IF B * 10 > M THEN GO TO 420
440 LET M = M - B * 10
450 PRINT "PRODUCTION IN 100S ?"
470 INPUT W
480 IF W > 10 * P THEN GO TO 470
490 LET I = I + W
500 LET S = S / (RND (3) + 3)
510 IF B < 6 THEN GO TO 550
520 LET S = S + (B - 6) * 5
530 LET B = 6
550 IF B < 2 THEN GO TO 600
560 LET S = S + (B - 2) * 7
570 LET B = 2
600 LET S = S + B * B
610 IF S > I THEN LET S = I
620 LET M = M + S * 10
630 LET I = I - S
640 LET M = M - 20 * P - 20
650 LET M = M - (W - 1) * 5
660 IF NOT M < 0 THEN GO TO 100
800 PRINT "YOU ARE BANKRUPT."
```

Sample Run

```
MONTH      1
YOU HAVE          $90000
             1 PLANTS
             000 WIDGETS
LAST MONTH YOU SOLD   1000
AD BUDGET IN 1000S?


MONTH     29
YOU HAVE          $101000
             1 PLANTS
             000 WIDGETS
LAST MONTH YOU SOLD  1300
BUY A PLANT?


MONTH     61
YOU HAVE          $71000
             2 PLANTS
             200 WIDGETS
LAST MONTH YOU SOLD  1900
AD BUDGET IN 1000S?


MONTH     85
YOU HAVE          $50000
             3 PLANTS
             000 WIDGETS
LAST MONTH YOU SOLD 3600
AD BUDGET IN 1000S?


MONTH    121
YOU HAVE          $53000
             5 PLANTS
             600 WIDGETS
LAST MONTH YOU SOLD 5300
AD BUDGET IN 10000S?
```

Sinclair ZX81 & Timex/Sinclair 1000

```
  5 RAND
  6 LET Z=1
 10 LET P=1
 20 LET M=900
 30 LET I=0
 40 LET S=10
100 CLS
110 PRINT "MONTH  ";Z
120 LET Z=Z+1
300 PRINT "YOU HAVE";TAB 15;"$"
;M;"00"
310 PRINT TAB 8;P;" PLANTS"
320 PRINT TAB 8;I;"00 WIDGETS"
330 PRINT "LAST MONTH YOU SOLD
";S;"00"
350 IF M<1000 THEN GOTO 410
360 PRINT "BUY A PLANT?"
370 INPUT Y$
380 IF Y$<>"Y" THEN GOTO 410
390 LET P=P+1
400 LET M=M-1000
410 PRINT "AD BUDGET IN 1000S ?
"
420 INPUT B
430 IF B*10>M THEN GOTO 420
440 LET M=M-B*10
450 PRINT "PRODUCTION IN 100S ?
"
470 INPUT W
480 IF W>10*P THEN GOTO 470
490 LET I=I+W
500 LET S=INT (S/INT (RND*3+3))
510 IF B<6 THEN GOTO 550
520 LET S=S+(B-6)*5
530 LET B=6
550 IF B<2 THEN GOTO 600
560 LET S=S+(B-2)*7
570 LET B=2
600 LET S=S+B*B
610 IF S>I THEN LET S=I
620 LET M=M+S*10
630 LET I=I-S
640 LET M=M-20*P-20
650 LET M=M-(W-1)*5
660 IF M>=0 THEN GOTO 100
800 PRINT "YOU ARE BANKRUPT."
```

Oil Company

Mark Phaneuf

This program puts the player in charge of a petroleum company. Through investment, your profits are increased or decreased. The program, written in DEC BASIC-OLUS for a PDP-11/50, is lengthy but is easily run and challenging to win.

Mark Phaneuf, 81 Tulsa St., Springfield, MA 01118

Some notes on the BASIC-PLUS functions:

If no verb is stated before the statement, LET is assumed.

Multiple statements are used; there is more than one statement per line.

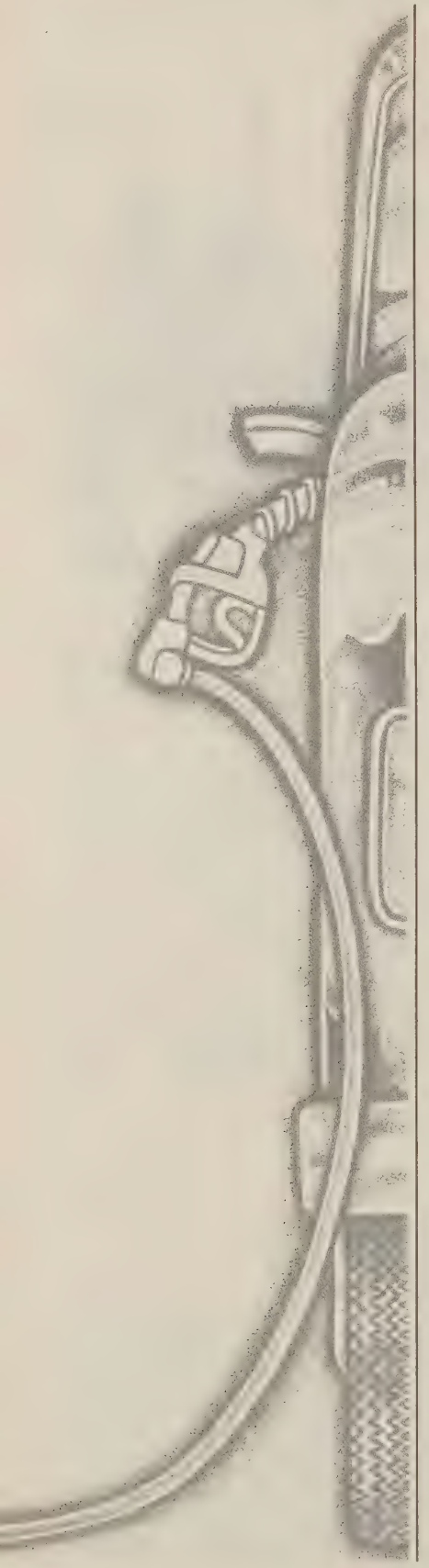
A formating procedure is used to print multiple statements. It prints the first statement beside the line number and the remaining statements below, as in Line 190.

The "&" = Print.

Statements such as in Line 280 are print-using statements. Line 280 will print "P" inserting commas "," and a decimal point "." and rounding off.

Sample Run

```
YOU HAVE JUST INHERITED AN OIL COMPANY. THIS OIL COMPANY WAS
LEFT TO YOU IN A WILL OF A DISTANT UNCLE. HE WISHED THAT
YOU WOULD INVEST HIS MONEY WISELY SO THAT YOU COULD KEEP
THE OIL COMPANY IN THE FAMILY. DON'T DISAPPOINT YOUR UNCLE
BY MAKING BAD INVESTMENTS
DO YOU WANT INSTRUCTIONS? N
HOW MANY YEARS DO YOU WANT TO PLAY (5-100)? 25
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 1
        ASSETS          $50,000.00
        TOTAL NUMBER OF GALLONS OF CRUDE OIL        2,000.00
        TOTAL # OF REFINERIES 1
        TOTAL # OF WELLS OWNED 1
COMMAND:4
REFINING COMMAND:2
        GASOLINE            $0.76
        FUEL OIL            $0.49
        OIL PRODUCTS        $0.61
        KEROSENE            $0.62
        LUBRICANTS          $0.86
REFINING COMMAND:1
WHAT DO YOU WANT TO REFINE? F
THE REFINING PRICE OF FUEL OIL IS  $0.49
HOW MUCH FUEL OIL TO REFINE? 2000
REFINING COMMAND:4
ROYALTIES COME IN!!!!
YOU RECEIVE          726.00 DOLLARS
YOU RECEIVE          871.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 2
        ASSETS       $49,749.60
        TOTAL NUMBER OF GALLONS OF CRUDE OIL         871.00
        TOTAL # OF REFINERIES 1
        TOTAL # OF WELLS OWNED 1
COMMAND:1
 THE COST OF IMPORTED OIL IS  $19.85 FOR 50 GALLONS
HOW MANY BARRELS DO YOU WANT TO BUY? 100
ROYALTIES COME IN!!!!
YOU RECEIVE          602.00 DOLLARS
YOU RECEIVE          974.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 3
        ASSETS       $48,366.40
        TOTAL NUMBER OF GALLONS OF CRUDE OIL        6,845.00
        TOTAL # OF REFINERIES 1
        TOTAL # OF WELLS OWNED 1
COMMAND:2
EXPORTING COMMAND:2
THE PRODUCTS IN STOCK ARE:
        GASOLINE                        0.00 GALLONS
        FUEL OIL                    1,940.00 GALLONS
        OIL PRODUCTS                    0.00 GALLONS
        KEROSENE                        0.00 GALLONS
        LUBRICANTS                      0.00 GALLONS
EXPORTING COMMAND:3
        GASOLINE            $1.09
        FUEL OIL            $0.42
        OIL PRODUCTS        $0.72
        KEROSENE            $0.85
        LUBRICANTS          $0.70
EXPORTING COMMAND:3
        GASOLINE            $1.09
        FUEL OIL            $0.42
        OIL PRODUCTS        $0.72
        KEROSENE            $0.85
        LUBRICANTS          $0.70
EXPORTING COMMAND:1
WHAT DO YOU WANT TO EXPORT? F
THE EXPORTING PRICE OF FUEL OIL IS   $0.42
HOW MUCH FUEL OIL TO EXPORT? 1900
EXPORTING COMMAND:4
ROYALTIES COME IN!!!!
YOU RECEIVE          923.00 DOLLARS
YOU RECEIVE          722.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 4
        ASSETS       $50,095.00
        TOTAL NUMBER OF GALLONS OF CRUDE OIL        7,567.00
        TOTAL # OF REFINERIES 1
        TOTAL # OF WELLS OWNED 1
COMMAND:3
THE PRICE OF A REFINERY IS  $31,244.80
DO YOU WANT TO BUILD A REFINERY? YES
HOW MANY REFINERIES TO BUILD? 1
STORAGE FIRE! HALF OF ALL PRODUCTS IN STORAGE ARE DESTROYED
ROYALTIES COME IN!!!!
YOU RECEIVE          554.00 DOLLARS
YOU RECEIVE          499.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
```



```
                    YEAR 5
          ASSETS        $19,404.20
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       8,066.00
          TOTAL # OF REFINERIES 2
          TOTAL # OF WELLS OWNED 1
COMMAND:7
WHERE DO YOU WANT TO DRILL? 2,2
NO OIL FOUND
DO YOU WANT TO DRILL AGAIN? Y
WHERE DO YOU WANT TO DRILL? 3,4
NO OIL FOUND
DO YOU WANT TO DRILL AGAIN? 6,7
DO YOU WANT TO DRILL AGAIN? Y
WHERE DO YOU WANT TO DRILL? 6,7
OIL FOUND AT 6,000 FEET
DO YOU WANT TO DRILL AGAIN? Y
WHERE DO YOU WANT TO DRILL? 8,1
OIL FOUND AT 2,000 FEET
YOU RAN OUT OF MONEY
ROYALTIES COME IN!!!!
YOU RECEIVE          1,208.00 DOLLARS
YOU RECEIVE          1,336.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 6
          ASSETS        $2,612.18
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       9,402.00
          TOTAL # OF REFINERIES 2
          TOTAL # OF WELLS OWNED 2
COMMAND:4
REFINING COMMAND:2 ''
          GASOLINE             $0.50
          FUEL OIL             $0.56
          OIL PRODUCTS         $0.61
          KEROSENE             $0.65
          LUBRICANTS           $0.83
REFINING COMMAND:1
WHAT DO YOU WANT TO REFINE? G
THE REFINING PRICE OF GASOLINE IS  $0.50
HOW MUCH GASOLINE TO REFINE? 4000
REFINING COMMAND:4
ROYALTIES COME IN!!!!
YOU RECEIVE          1,702.00 DOLLARS
YOU RECEIVE          1,416.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 7
          ASSETS        $2,297.45
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       6,818.00
          TOTAL # OF REFINERIES 2
          TOTAL # OF WELLS OWNED 2
COMMAND:5
          GASOLINE                  3,880
          FUEL OIL                     20
          OIL PRODUCTS                  0
          KEROSENE                      0
          LUBRICANTS                    0
COMMAND:6
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 2  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 3  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 6  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0
 7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
COMMAND:8
DO YOU WANT TO SELL REFINERIES OR WELLS? R
 THE SELLING PRICE OF A REFINERY IS  $20,038.60
HOW MANY REFINERIES TO SELL? 1
ROYALTIES COME IN!!!!
YOU RECEIVE          1,670.00 DOLLARS
YOU RECEIVE            748.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 8
          ASSETS       $24,006.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       7,566.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 2
COMMAND:11 F
ROYALTIES COME IN!!!!
YOU RECEIVE            656.00 DOLLARS
YOU RECEIVE          1,580.00 GALLONS OF CRUDE OIL
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 9
          ASSETS       $24,662.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       9,146.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 2
COMMAND:5
          GASOLINE                  3,880
          FUEL OIL                     20
          OIL PRODUCTS                  0
          KEROSENE                      0
          LUBRICANTS                    0
COMMAND:9
<#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#> <#>
                    YEAR 9
          ASSETS       $24,662.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       9,146.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 2
COMMAND:10
SURE??????? YES
++++ SOLD OUT FOR      50,716.60 DOLLARS ++++
********************* OWNING CERTIFICATE *********************
YEARS LASTED 9     WELLS OWNED  2
ASSETS                      $75,378.60 DOLLARS
CRUDE OIL                     9,146.00 GALLONS
REFINERIES                        1.00
GASOLINE                      3,880.00 GALLONS
FUEL OIL                         20.00 GALLONS
OIL PRODUCTS                      0.00 GALLONS
KEROSENE                          0.00 GALLONS
LUBRICANTS                        0.00 GALLONS
RATING (1 BAD) - (10 PERFECT) 0
```

Program Listing

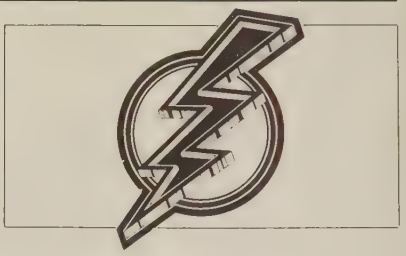
```
1 ! SAVE, KNGOIL.BAS, PHANEUF, KIM, KIM1
10 ON ERROR GOTO 4850
20 OPEN 'KB:' AS FILE 1
30 $"YOU HAVE JUST INHERITED AN OIL COMPANY. THIS OIL COMPANY WAS"
40 $"LEFT TO YOU IN A WILL OF A DISTANT UNCLE. HE WISHED THAT "
50 $"YOU WOULD INVEST HIS MONEY WISELY SO THAT YOU COULD KEEP"
60 $"THE OIL COMPANY IN THE FAMILY. DON'T EISAPPOINT YOUR UNCLE"
70 $"BY MAKING BAD INVESTMENTS"
80 DIM Y%(10,20),Z%(10,20)
90 RANDOMIZE
100 Y1=0
110 FOR I=0 TO 10
120 FOR J=0 TO 20
130 X=INT(5-1)*RND(1)+1
140 Y%(I,J)=X
150 NEXT J
160 NEXT I
170 Y1=Y1+1
180 IF Y1=200 THEN 190  ELSE 130
190 D=1:
        P=50000:
        C=2000:
        R=1:
        G=0:
        F=0:
        K=0:
        L=0:
        O=0:
        W=1:
        X1=0
200 INPUT 'DO YOU WANT INSTRUCTIONS'$I$
210 IF LEFT(I$,1)='Y' THEN 3390
220 INPUT 'HOW MANY YEARS DO YOU WANT TO PLAY (5-100)'$B5
230 B5=INT(B5)
240 IF B5<5 OR B5>100 THEN 220
250 IF P<=0 THEN 3050
260 $"<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>"
270 $"                YEAR"$D
280 $TAB(10)$:
        $ USING 'ASSETS $$$$$,$$$,$$$.$$',P
290 $TAB(10)$:
        $ USING 'TOTAL NUMBER OF GALLONS OF CRUDE OIL $$$,$$$,$$$.$$',C
300 $TAB(10)$:
        $"TOTAL $ OF REFINERIES"$R
310 $TAB(10)$:
        $"TOTAL $ OF WELLS OWNED"$W
320 INPUT #1,'COMMAND:'$Z
330 IF Z<1 OR Z>11 OR Z<>INT(Z) THEN 340  ELSE 400
340 $"COMMANDS ARE:"
350 $"1-IMPORT OIL,2-EXPORT OIL PRODUCTS,3-BUILD REFINERIES"
360 $"4-REFINE OIL,5-STORAGE REPORT,6- OIL FIELD DATA"
370 $"7-DRILL WELLS, 8-SELL WELLS OR REFINERIES, 9-OWNERS STATUS, 10-SELL OUT
380 $"11-PASS YEAR"
390 GOTO 320
400 ON Z GOTO 410,500,1160,1320,2280,2340,2380,4650,4830,3000,4840
410 RANDOMIZE
420 B=INT(20-10)*RND(1)+10
430 $ USING ' THE COST OF IMPORTED OIL IS $$$$.$$ FOR 50 GALLONS',B
440 INPUT 'HOW MANY BARRELS DO YOU WANT TO BUY'$F7
450 F7=INT(F7)
460 IF C<0 THEN 320
470 IF F7*B>P THEN 480  ELSE 490
480 $ USING 'YOU ONLY HAVE   $$$,$$$,$$$.$$ DOLLARS',P:
        GOTO 440
490 C=C+(F7*50):
        P=P-(F7*B):
        GOTO 2720
500 RANDOMIZE
510 O1=(1.25-.38)*RND(1)+.38
520 F1=(1-.33)*RND(1)+.33
530 O1=(.95-.48)*RND(1)+.48
540 K1=(.95-.48)*RND(1)+.48
550 L1=(1.25-.58)*RND(1)+.58
560 INPUT #1,'EXPORTING COMMAND:'$E
570 IF E<1 OR E>4 OR E<>INT(E) THEN 580  ELSE 620
580 $"EXPORTING COMMANDS ARE:"
590 $"1-EXPORT OIL PRODUCTS,2-PRODUCTS IN STOCK"
600 $"3-EXPORTING PRICE LIST, 4-DONE EXPORTING"
610 GOTO 560
620 ON E GOTO 630,1020,1090,1150
630 INPUT 'WHAT DO YOU WANT TO EXPORT'$E$
640 IF LEFT(E$,1)='O' THEN 650  ELSE 710
650 $ USING 'THE EXPORTING PRICE OF GASOLINE IS $$$.$$',G1
660 INPUT 'HOW MUCH GASOLINE TO EXPORT'$G2
670 IF G2<=0 THEN 560
680 IF G2>O THEN 690  ELSE 700
690 $ USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF GASOLINE',G:
        GOTO 660
700 G=G-G2:
        P=P+(G2*G1):
        GOTO 560
710 IF LEFT(E$,1)='F' THEN 720  ELSE 780
720 $ USING 'THE EXPORTING PRICE OF FUEL OIL IS  $$$.$$',F1
730 INPUT 'HOW MUCH FUEL OIL TO EXPORT'$F2
740 IF F2<=0 THEN 560
750 IF F2>F THEN 760  ELSE 770
760 $ USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF FUEL OIL',F:
        GOTO 730
770 F=F-F2:
        P=P+(F2*F1):
        GOTO 560
780 IF LEFT(E$,1)='O' THEN 790  ELSE 850
790 $ USING 'THE EXPORTING PRICE OF OIL PRODUCTS IS $$$.$$',O1
800 INPUT 'HOW MUCH OIL PRODUCTS TO EXPORT'$O2
810 IF O2<=0 THEN 560
820 IF O2>O THEN 830  ELSE 840
830 $ USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF OIL PRODUCTS',O:
        GOTO 800
840 O=O-O2:
        P=P+(O2*O1):
        GOTO 560
850 IF LEFT(E$,1)='K' THEN 860  ELSE 920
860 $ USING 'THE EXPORTING PRICE OF KEROSENE IS $$$.$$',K1
870 INPUT 'HOW MUCH KEROSENE TO EXPORT'$K2
880 IF K2<=0 THEN 560
890 IF K2>K THEN 900  ELSE 910
900 $ USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF KEROSENE',K:
        GOTO 870
910 K=K-K2:
        P=P+(K2*K1):
        GOTO 560
920 IF LEFT(E$,1)='L' THEN 930  ELSE 1000
930 $ USING 'THE EXPORTING PRICE OF LUBRICANTS IS $$$.$$',L1
940 INPUT 'HOW MUCH LUBRICANTS TO EXPORT'$L2
950 IF L2<=0 THEN 560
960 IF L2>L THEN 970  ELSE 980
970 $ USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF LUBRICANTS',L:
        GOTO 940
980 L=L-L2:
        P=P+(L2*L1):
        GOTO 560
990 $"THE PRODUCTS YOU CAN EXPORT ARE:"
1000 $"GASOLINE,FUEL OIL, OIL PRODUCTS , KEROSENE, AND LUBRICANTS"
1010 GOTO 560
1020 $"THE PRODUCTS IN STOCK ARE:"
1030 $TAB(10)$:
        $ USING 'GASOLINE              $$$,$$$,$$$.$$ GALLONS',G
1040 $TAB(10)$:
        $ USING 'FUEL OIL              $$$,$$$,$$$.$$ GALLONS',F
1050 $TAB(10)$:
        $ USING 'OIL PRODUCTS          $$$,$$$,$$$.$$ GALLONS',O
1060 $TAB(10)$:
        $ USING 'KEROSENE              $$$,$$$,$$$.$$ GALLONS',K
1070 $TAB(10)$:
        $ USING 'LUBRICANTS            $$$,$$$,$$$.$$ GALLONS',L
1080 GOTO 560
1090 $TAB(10)$:
        $ USING 'GASOLINE              $$$.$$',G1
1100 $TAB(10)$:
        $ USING 'FUEL OIL              $$$.$$',F1
1110 $TAB(10)$:
        $ USING 'OIL PRODUCTS          $$$.$$',O1
1120 $TAB(10)$:
        $ USING 'KEROSENE              $$$.$$',K1
1130 $TAB(10)$:
        $ USING 'LUBRICANTS            $$$.$$',L1
1140 GOTO 560
1150 GOTO 2720
1160 RANDOMIZE
1170 R1=INT(50000-30000)*RND(1)+30000
1180 $ USING 'THE PRICE OF A REFINERY IS $$$$,$$$.$$',R1
1190 INPUT 'DO YOU WANT TO BUILD A REFINERY'$R$
1200 IF LEFT(R$,1)='Y' THEN 1240  ELSE 1210
1210 IF LEFT(R$,1)='N' THEN 320  ELSE 1220
1220 $"  I NEED (YES OR NO)"
1230 GOTO 1190
1240 INPUT 'HOW MANY REFINERIES TO BUILD'$R2
1250 IF R2<0 OR R2<>INT(R2) THEN 1260  ELSE 1280
1260 $"YOU HAVE TO BUY EITHER 0 OR ANY WHOLE NUMBER OF REFINERIES"
1270 GOTO 1240
1280 IF R2*R1>P THEN 1290  ELSE 1310
1290 $ USING 'YOU ONLY HAVE    $$$,$$$,$$$.$$ DOLLARS',P
1300 GOTO 1240
1310 P=P-(R2*R1):
        R=R+R2:
        GOTO 2720
1320 RANDOMIZE
1330 G3=(1-.5)*RND(1)+.5
1340 F3=(.75-.45)*RND(1)+.45
1350 O3=(.7-.6)*RND(1)+.6
1360 L3=(1-.7)*RND(1)+.7
1370 K3=(.7-.6)*RND(1)+.6
1380 INPUT #1,'REFINING COMMAND:'$R3
1390 IF R3<1 OR R3>4 OR R3<>INT(R3) THEN 1400  ELSE 1430
1400 $"THE REFINING COMMANDS ARE:"
1410 $"1-REFINE,2-REFINING PRICE LIST,3-COMPANY STATUS,4-DONE REFINING"
1420 GOTO 1380
1430 ON R3 GOTO 1440,2180,2240,2270
1440 INPUT 'WHAT DO YOU WANT TO REFINE'$A$
1450 IF LEFT(A$,1)='G' THEN 1460  ELSE 1590
1460 $ USING 'THE REFINING PRICE OF GASOLINE IS $$$.$$',G3
1470 INPUT 'HOW MUCH GASOLINE TO REFINE'$G4
1480 G4=INT(G4)
1490 IF G4<=0 THEN 1380
1500 IF G4>C THEN 1510  ELSE 1520
1510 $ USING 'YOU ONLY HAVE $$$,$$$,$$$ GALLONS OF CRUDE OIL',C:
        GOTO 1470
1520 IF G4/125000>R THEN 1530  ELSE 1540
1530 $"THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES":
        GOTO 1470
1540 IF G4*G3>P THEN 1550  ELSE 1560
1550 $"YOU DON'T HAVE ENOUGH MONEY":
        GOTO 1470
1560 IF (X1+G4)/125000>R THEN 1570  ELSE 1580
1570 $"YOU CAN REFINE"$(R*125000)-X1$"GALLONS":
        GOTO 1470
1580 G=G+(G4-(G4*.03)):
        G=INT(G):
        P=P-(G4*G3):
        C=C-G4:
        X1=X1+G4:
        GOTO 1380
```

```
1590 IF LEFT(A$,1)='F' THEN 1600  ELSE 1730
1600 $ USING 'THE REFINING PRICE OF FUEL OIL IS $$$.$$',F3
1610 INPUT 'HOW MUCH FUEL OIL TO REFINE'$F4
1620 F4=INT(F4)
1630 IF F4<=0 THEN 1380
1640 IF F4>C THEN 1650  ELSE 1660
1650 $ USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
     GOTO 1610
1660 IF F4/125000>R THEN 1670  ELSE 1680
1670 $'THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES':
     GOTO 1610
1680 IF F4*F3>P THEN 1690  ELSE 1700
1690 $'YOU DON'T HAVE ENOUGH MONEY':
     GOTO 1610
1700 IF (X1+F4)/125000>R THEN 1710  ELSE 1720
1710 $'YOU CAN REFINE'$(R*125000)-X1$'GALLONS':
     GOTO 1610
1720 F=F+(F4-(F4*.03)):
     F=INT(F):
     P=P-(F4*F3):
     C=C-F4:
     X1=X1+F4:
     GOTO 1380
1730 IF LEFT(A$,1)='O' THEN 1740  ELSE 1870
1740 $ USING ' THE REFINING PRICE OF OIL PRODUCTS IS $$$.$$',O3
1750 INPUT 'HOW MUCH OIL PRODUCTS TO REFINE'$O4
1760 O4=INT(O4)
1770 IF O4<=0 THEN 1380
1780 IF O4>C THEN 1790  ELSE 1800
1790 $ USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
     GOTO 1750
1800 IF O4/125000>R THEN 1810  ELSE 1820
1810 $'THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES':
     GOTO 1750
1820 IF O4*O3>P THEN 1830  ELSE 1840
1830 $'YOU DON'T HAVE ENOUGH MONEY':
     GOTO 1750
1840 IF (X1+O4)/125000>R THEN 1850  ELSE 1860
1850 $'YOU CAN REFINE'$(R*125000)-X1$'GALLONS':
     GOTO 1750
1860 O=O+(O4-(O4*.03)):
     O=INT(O):
     P=P-(O4*O3):
     C=C-O4:
     X1=X1+O4:
     GOTO 1380
1870 IF LEFT(A$,1)='K' THEN 1880  ELSE 2010
1880 $ USING ' THE REFINING PRICE OF KEROSENE IS $$$.$$',K3
1890 INPUT 'HOW MUCH KEROSENE TO REFINE'$K4
1900 K4=INT(K4)
1910 IF K4<=0 THEN 1380
1920 IF K4>C THEN 1930  ELSE 1940
1930 $ USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
     GOTO 1890
1940 IF K4/125000>R THEN 1950  ELSE 1960
1950 $'THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES':
     GOTO 1890
1960 IF K4*K3>P THEN 1970  ELSE 1980
1970 $'YOU DON'T HAVE ENOUGH MONEY':
     GOTO 1890
1980 IF (X1/K4)/125000>R THEN 1990  ELSE 2000
1990 $'YOU CAN REFINE'$(R*125000)-X1$'GALLONS':
     GOTO 1890
2000 K=K+(K4-(K4*.03)):
     K=INT(K):
     P=P-(K4*K3):
     C=C-K4:
     X1=X1+K4:
     GOTO 1380
2010 IF LEFT(A$,1)='L' THEN 2020  ELSE 2150
2020 $ USING ' THE REFINING PRICE OF LUBRICANTS IS $$$.$$',L3
2030 INPUT 'HOW MUCH LUBRICANTS TO REFINE'$L4
2040 L4=INT(L4)
2050 IF L4<=0 THEN 1380
2060 IF L4>C THEN 2070  ELSE 2080
2070 $ USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
     GOTO 2030
2080 IF L4/125000>R THEN 2090  ELSE 2100
2090 $'THAT IS TOO MUCH FOR YOUR REFINERIES ':
     GOTJ 2030
2100 IF L4*L3>P THEN 2110  ELSE 2120
2110 $'YOU DON'T HAVE ENOUGH MONEY':
     GOTO 2030
2120 IF (X1+L4)/125000>R THEN 2130  ELSE 2140
2130 $'YOU CAN REFINE'$(R*125000)-X1$'GALLONS':
     GOTO 2030
2140 L=L+(L4-(L4*.03)):
     L=INT(L):
     P=P-(L4*L3):
     C=C-L4:
     X1=X1+L4:
     GOTO 1380
2150 $'THE PRODUCTS THAT YOU CAN REFINE ARE:'
2160 $'GASOLINE, FUEL OIL, OIL PRODUCTS, KEROSENE, AND LUBRICANTS'
2170 GOTO 1440
2180 $TAB(10)$:
     $ USING 'GASOLINE            $$.$$',G3
2190 $TAB(10)$:
     $ USING 'FUEL OIL            $$.$$',F3
2200 $TAB(10)$:
     $ USING 'OIL PRODUCTS        $$.$$',O3
2210 $TAB(10)$:
     $ USING 'KEROSENE            $$.$$',K3
2220 $TAB(10)$:
     $ USING 'LUBRICANTS          $$.$$',L3
2230 GOTO 1380
2240 $ USING 'ASSETS              ###,###,###.$$ DOLLARS',P
2250 $ USING 'CRUDE OIL          ###,###,###.$$ GALLONS',C
2260 GOTO 1380
2270 GOTO 2720
2280 $TAB(10)$:
     $ USING 'GASOLINE            ###,###,###',G
2290 $TAB(10)$:
     $ USING 'FUEL OIL            ###,###,###',F
2300 $TAB(10)$:
     $ USING 'OIL PRODUCTS        ###,###,###',O
2310 $TAB(10)$:
     $ USING 'KEROSENE            ###,###,###',K
2320 $TAB(10)$:
     $ USING 'LUBRICANTS          ###,###,###',L
2330 GOTO 320
2340 $ USING '## ',J$ FOR J=0 TO 20:
     $:
     FOR I=1 TO 10
2350 $ USING '## ',I$:
     $ USING '## ',Z%(I,J)$ FOR J=1 TO 20
2360 $:
     NEXT I
2370 GOTO 320
2380 INPUT 'WHERE DO YOU WANT TO DRILL'$I,J
2390 IF I>10 OR J>20 OR I<1 OR J<1 THEN 2400  ELSE 2410
2400 $'I NEED 2 WHOLE NUMBERS THE 1ST(1-10) THEN 2ND(1-20)':
     GOTO 2380
2410 I=INT(I):
     J=INT(J):
     IF W9=200 THEN 2990
2420 ON Y%(I,J) GOTO 2590,2540,2490,2440,2430
2430 $'YOU ALREADY DRILLED THERE':
     GOTO 2380
2440 $'OIL FOUND AT 2,000 FEET':
     P=P-2000
2450 IF P<0 THEN 2460  ELSE 2470
2460 P=P+2000:
     $'YOU RAN OUT OF MONEY':
     GOTO 2720
2470 Y%(I,J)=5
2480 Z%(I,J)=5:
     GOTO 2640
2490 $'OIL FOUND AT 4,000 FEET':
     P=P-4000
2500 IF P<0 THEN 2510  ELSE 2520
2510 P=P+4000:
     $'YOU RAN OUT OF MONEY':
     GOTO 2720
2520 Y%(I,J)=5
2530 Z%(I,J)=5:
     GOTO 2640
2540 $'OIL FOUND AT 6,000 FEET':
     P=P-6000
2550 IF P<0 THEN 2560  ELSE 2570
2560 P=P+6000:
     $'YOU RAN OUT OF MONEY':
     GOTO 2720
2570 Y%(I,J)=5
2580 Z%(I,J)=5:
     GOTO 2640
2590 $'NO OIL FOUND':
     P=P-6000
2600 IF P<0 THEN 2610  ELSE 2620
2610 P=P+6000:
     $'YOU RAN OUT OF MONEY':
     GOTO 2720
2620 Y%(I,J)=5
2630 Z%(I,J)=5:
     GOTO 2650
2640 W=W+1
2650 W9=W9+1
2660 INPUT 'DO YOU WANT TO DRILL AGAIN'$D$
2670 IF D$='Y' OR D$='N' THEN 2680  ELSE 2660
2680 IF LEFT(D$,1)='Y' THEN 2380
2690 GOTO 2720
2700 IF LEFT(Y9$,1)='1' THEN 2720  ELSE 2710
2710 IF LEFT(Y9$,1)='2' THEN 320  ELSE 2690
2720 RANDOMIZE
2730 U=INT(60*RND(1))
2740 IF U=50 THEN 2750  ELSE 2760
2750 R=R*.75:
     R=INT(R):
     GOTO 2780
2760 IF U=40 THEN 2770  ELSE 2810
2770 P=P*.5:
     P=INT(P):
     GOTO 2790
2780 $'REFINERIES EXPLODE'$R$'REFINERIES LEFT':
     GOTO 2890
2790 $ USING 'BAD INVESTMENT   ###,###,###.## DOLLARS LEFT',P
2800 GOTO 2890
2810 IF U=30 THEN 2820  ELSE 2840
2820 $ USING 'OIL SPILL! ###,###,### GALLONS OF CRUDE OIL DESTROYED',C*.75
2830 C=C-(C*.75):
     GOTO 2890
2840 IF U=20 THEN 2850  ELSE 2860
2850 $'FIRE DAMAGE!'$W*.5$'WELLS DESTROYED':
     W=W-(W*.5):
     GOTO 2890
2860 IF U=10 THEN 2870  ELSE 2890
2870 $'STORAGE FIRE! HALF OF ALL PRODUCTS IN STORAGE ARE DESTROYED'
2880 G=G/2:
     F=F/2:
     L=L/2:
     K=K/2:
     O=O/2
2890 D=D+1:
     X1=0
2900 IF W>0 THEN $'ROYALTIES COME IN!!!!'  ELSE 2970
2910 RANDOMIZE
2920 W1=INT(1000-100)*RND(1)+100:
     C5=INT(1000-100)*RND(1)+100
2930 W1=INT(W1):
     C5=INT(C5)
2940 P=P+(W*W1):
     C=C+(C5*W)
2950 $ USING 'YOU RECEIVE   ###,###,###.## DOLLARS',W*W1
2960 $ USING 'YOU RECEIVE   ###,###,###.## GALLONS OF CRUDE OIL',C5*W
2970 IF W9=200 OR D=85 OR P=0 THEN 2990  ELSE 2980
2980 G=INT(G):
     F=INT(F):
     O=INT(O):
     K=INT(K):
     L=INT(L):
     W=INT(W):
     GOTO 250
```

```
2990 $"GAME OVER":
     GOTO 3050
3000 INPUT 'SURE?????';M9$
3010 IF M9$='YES' THEN 3030
3020 IF M9$='NO' THEN 320
     ELSE $" YES OR NO":
     GOTO 3000
3030 $ USING '++++ SOLD OUT FOR $$$,$$$,$$$.$$ DOLLARS ++++';((W+P+C+R)*1.
3040 P=P+((W+P+C+R)*1.5)
3050 $"********************* OWNING CERTIFICATE **********************"
3060 $"YEARS LASTED";D;"    WELLS OWNED ";W
3070 $ USING 'ASSETS                $$$$$,$$$,$$$.$$ DOLLARS',P
3080 $ USING 'CRUDE OIL               $$$,$$$,$$$.$$ GALLONS',C
3090 $ USING 'REFINERIES              $$$,$$$,$$$.$$',R
3100 $ USING 'GASOLINE                $$$,$$$,$$$.$$ GALLONS',G
3110 $ USING 'FUEL OIL                $$$,$$$,$$$.$$ GALLONS',F
3120 $ USING 'OIL PRODUCTS            $$$,$$$,$$$.$$ GALLONS',O
3130 $ USING 'KEROSENE                $$$,$$$,$$$.$$ GALLONS',K
3140 $ USING 'LUBRICANTS              $$$,$$$,$$$.$$ GALLONS',L
3150 Q3=(((P+C)/5000)+((W+R)*.1))*(D-1))
3160 IF Q3>=486.5 THEN 3170  ELSE 3180
3170 R9=10:
     GOTO 3370
3180 IF Q3>=437.85 THEN 3190  ELSE 3200
3190 R9=9:
     GOTO 3370
3200 IF Q3>=389.2 THEN 3210  ELSE 3220
3210 R9=8:
     GOTO 3370
3220 IF Q3>=340.55 THEN 3230  ELSE 3240
3230 R9=7:
     GOTO 3370
3240 IF Q3>=291.9 THEN 3250  ELSE 3260
3250 R9=6:
     GOTO 3370
3260 IF Q3>=243.25 THEN 3270  ELSE 3280
3270 R9=5:
     GOTO 3370
3280 IF Q3>=194.6 THEN 3290  ELSE 3300
3290 R9=4:
     GOTO 3370
3300 IF Q3>=145.95 THEN 3310  ELSE 3320
3310 R9=3:
     GOTO 3370
3320 IF Q3>=97.3 THEN 3330  ELSE 3340
3330 R9=2:
     GOTO 3370
3340 IF Q3>=48.65 THEN 3350  ELSE 3360
3350 R9=1:
     GOTO 3370
3360 R9=0:
     GOTO 3370
3370 $"RATING (1 BAD) - (10 PERFECT)";R9
3380 GOTO 4860
3390 $"                    COMMAND-1"
3400 $" ALLOWS YOU TO PURCHASE IMPORTED CRUDE OIL AT A PRICE BETWEEN"
3410 $"$10.00 AND $20.00 PER FIFTY GALLON BARREL"
3420 $:
     $:
     $
3430 $"                    COMMAND-2"
3440 $" ALLOWS YOU TO EXPORT YOUR REFINED PRODUCTS NAMELY;"
3450 $" GASOLINE SELLING FOR A PRICE BETWEEN $.38 AND $1.25 P.G."
3460 $" FUEL OIL SELLING FOR A PRICE BETWEEN $.33 AND $1.00 P.G."
3470 $" OIL PRODUCTS SELLING FOR A PRICE BETWEEN $.48 AND $.95 P.G."
3480 $" KEROSENE SELLING FOR A PRICE BETWEEN $.48 AND $.95 P.G."
3490 $" LUBRICANTS SELLING FOR A PRICE BETWEEN $.58 AND $1.25 P.G."
3500 $
3510 $"                    SUGGESTIONS"
3520 $" CONSIDERATION SHOULD BE GIVEN TO EXPORTING A PRODUCT WHEN"
3530 $" ITS PRICE IS HIGH"
3540 $:
     $
3550 $" WHEN ASKED FOR AN EXPORTING COMMAND YOU MUST RESPOND WITH;"
3560 $
3570 $"          1-EXPORT"
3580 $" ALLOWS YOU TO EXPORT ANY REFINED PRODUCTS IN STOCK IN ANY GIVEN"
3590 $" QUANTITY AT THE SPECIFIED PRICE"
3600 $
3610 $"          2-PRODUCTS IN STOCK"
3620 $" GIVES YOU A LISTING OF YOUR REFINED PRODUCTS AND THE QUANTITY"
3630 $" OF EACH"
3640 $
3650 $"          3-EXPORTING PRICE LIST"
3660 $" GIVES YOU A LISTING OF ALL EXPORTABLE PRODUCTS AND THEIR PRICES"
3670 $
3680 $"          4-DONE EXPORTING"
3690 $" SIGNIFIES COMPLETION OF  COMMAND 2"
3700 $:
     $:
     $
3710 $"                    COMMAND-3"
3720 $" ALLOWS YOU TO BUILD REFINERIES. THE COST OF A REFINERY"
3730 $"RANGES FROM $30,000 TO $50,000 EACH"
3740 $:
     $:
     $
3750 $"                    COMMAND-4"
3760 $" ALLOWS YOU TO REFINE YOUR CRUDE OIL INTO THE FOLLOWING PRODUCTS"
3770 $" GASOLINE COSTING $.50 TO $1.00 P.G. TO REFINE"
3780 $" FUEL OIL COSTING $.45 TO $.75 P.G. TO REFINE"
3790 $" OIL PRODUCTS COSTING $.60 TO $.70 P.G. TO REFINE"
3800 $" KEROSENE COSTING $.60 TO $.70 P.G. TO REFINE"
3810 $" LUBRICANTS COSTING $.70 TO $1.00 P.G. TO REFINE"
3820 $:
     $
3830 $" WHEN ASKED FOR A REFINING COMMAND YOU MUST RESPOND WITH;"
3840 $"          1-REFINE"
3850 $" ALLOWS YOU TO REFINE CRUDE OIL INTO BY-PRODUCTS"
3860 $
3870 $"          2-REFINING PRICE LIST"
3880 $" GIVES YOU A LISTING OF ALL REFINABLE PRODUCTS AND THEIR PRICES"
3890 $
3900 $"          3-COMPANY STATUS"
3910 $
3920 $" TELLS YOU HOW MUCH MONEY AND CRUDE OIL YOU HAVE LEFT"
3930 $
3940 $"          4-DONE REFINING"
3950 $" SIGNIFIES COMPLETION OF COMMAND 4"
3960 $!
     $
3970 $"                    STIPULATIONS"
3980 $"          A- YOU ARE UNABLE TO REFINE MORE OF ANY PRODUCT"
3990 $" THAN YOU HAVE CRUDE OIL"
4000 $
4010 $"          B- A REFINERY IS LIMITED TO PRODUCING 125,000 GALLONS"
4020 $" PER YEAR. THEREFORE YOU ARE UNABLE TO REFINE MORE CRUDE OIL"
4030 $" THAN YOUR REFINERIES ARE CAPIBLE OF REFINING"
4040 $
4050 $"          C-THE GALLONAGE TIMES THE REFINING PRICE CAN'T"
4060 $" BE GRATER THAN YOUR ASSETS"
4070 $"                    STIPULATIONS"
4080 $" WHEN REFINING 3% OF THE CRUDE OIL THAT YOU WANTED TO REFINE"
4090 $" IS LOST IN THE REFINING PROCESS"
4100 $!
     $:
     $
4110 $"                    COMMAND-5
4120 $" GIVES A LISTING OF ALL YOUR REFINED PRODUCTS AND THE QUANTITY"
4130 $" IN WHICH THEY WERE REFINED"
4140 $!
     $:
     $
4150 $"                    COMMAND-6"
4160 $"GIVES YOU A MAP OF YOUR OIL FIELD. WHERE THERE IS A 'S'"
4170 $" YOU HAVE ALREADY DRILLED AND WHERE THERE IS A 'O' YOU"
4180 $" HAVE NOT DRILL YET"
4190 $!
     $:
     $
4200 $"                    COMMAND-7"
4210 $!" ALLOWS YOU TO DRILL WELLS IN A LOCATION OF YOUR CHOICE"
4220 $" YOU STATE WHERE YOU WANT TO DRILL BY PICKING TWO NUMBERS"
4230 $" THE FIRST(1-10) THEN SECOND(1-20). THESE NUMBERS MUST BE"
4240 $" SEPERATED BY A COMMA.          FOR EXAMPLE:"
4250 $" 1,1          5,5          10,20"
4260 $" IF OIL IS FOUND AT 2,000 FEET THEN THE COST FOR DRILLING IS"
4270 $" $2,000."
4280 $" IF OIL IS FOUND AT 4,000 FEET THEN THE COST FOR DRILLING IS"
4290 $" $4,000."
4300 $" IF OIL IS FOUND AT 6,000 FEET THEN THE COST FOR DRILLING IS"
4310 $" $6,000."
4320 $" IF NO OIL IS FOUND THEN THE COST FOR DRILLING IS $6,000"
4330 $" BUT YOU WILL NOT RECEIVE ANY WELL"
4340 $!
     $:
     $
4350 $"                    COMMAND-8"
4360 $" ALLOWS YOU TO SELL WELL AND REFINERIES"
4370 $" THE SELLING PRICE OF A WELL RANGES FROM $500.00 TO $5,000.00"
4380 $" DOLLARS EACH. THE SELLING PRICE OF A REFINERY RANGES FROM"
4390 $"$20,000 TO $30,000 EACH.  YOU MAY SELL A FEW OR ALL"
4400 $" YOU HAVE"
4410 $
4420 $"                    COMMAND-9"
4430 $" GIVES YOU AN OWNERS STATUS WHICH MEANS IT WILL TELL YOU YOUR"
4440 $" TOTAL ASSETS, TOTAL GALLONAGE OF CRUDE OIL, AMOUNT OF REFINERIES"
4450 $" OWNED, THE TOTAL NUMBER OF WELLS OWNED AND ALSO WHAT YEAR IT IS"
4460 $!
     $
4470 $"                    COMMAND-10"
4480 $" END THE GAME"
4490 $!
     $
4491 $"                    COMMAND-11"
4492 $" ALLOWS YOU TO PASS THE YEAR WHICH ALLOWS YOU TO COLLECT ROYALTIES"
4493 $" ON YOU EXISTING WELLS"
4500 $"                    ROYALTIES"
4510 $" EVERY YEAR ROYALTIES ARE PAID ON EXISTING WELLS."
4520 $"          A-YOU WILL RECEIVE BETWEEN 100 AND 1000"
4530 $" GALLONS OF CRUDE OIL PER WELL."
4540 $
4550 $"          B-YOU WILL RECEIVE BETWEEN 100 AND 1000"
4560 $" DOLLARS PER WELL"
4570 $
4580 $" EVERY YEAR YOU HAVE A CHANCE OF 1:15 TO GET A DISASTER"
4590 $" LISTED BELOW;"
4600 $"          A-FIRE DAMAGE";
     $"HALF OF ALL WELLS ARE DESTROYED"
4610 $"          B-STORAGE FIRE";
     $"HALF OF ALL PRODUCTS  IN STOCK ARE DESTROYED"
4620 $"          C-BAD INVESTMENT";
     $"HALF OF ALL ASSETS ARE LOST"
4630 $"          D-REFINERY EXPLOSION";
     $"75% OF ALL REFINERIES ARE LOST"
4640 GOTO 220
4650 INPUT 'DO YOU WANT TO SELL REFINERIES OR WELLS';Z5$
4660 Z5$=LEFT(Z5$,1):
     IF Z5$='R' THEN 4680
     ELSE IF Z5$='W' THEN 4750
      ELSE 320
4670 IF LEFT(Z5$,1)='W' THEN 4750  ELSE 4650
4680 RANDOMIZE
4690 X3=INT(30000-20000)*RND(1)+20000
4700 $ USING ' THE SELLING PRICE OF A REFINERY IS $$$$,$$$.$$',X3
4710 INPUT 'HOW MANY REFINERIES TO SELL';H
4720 IF H>R THEN $"YOU HAVE";R;"REFINERIES":
     GOTO 4700
4730 P=P+(H*X3)
4740 R=R-H:
     GOTO 2720
4750 RANDOMIZE
4760 X5=INT(5000-500)*RND(1)+500
4770 $"THE SELLING PRICE OF A REFINERY IS ";X5
4780 INPUT 'HOW MANY WELLS TO SELL';H1
4790 IF H1>W THEN $"YOU HAVE";W;"WELLS":
     GOTO 4770
4800 P=P+(X5*H1)
4810 W=W-H1:
     GOTO 2720
4820 GOTO 260
4830 GOTO 260
4840 GOTO 2720
4850 RESUME
4860 END
```

The Electric Company

Ralph White



Game Description

THE ELECTRIC COMPANY is an economic game/simulation. The user is made president of the company for ten years — unless mismanagement forces earlier retirement.

At the beginning of year one, the company has one million dollars, 1600 tons of coal, 2500 barrels of fuel oil, one power plant and 700 customers. All power plants have a capacity of one million kilowatts. The user may choose which type of fuel to use at each power plant he owns — the one he starts with as well as each one that is built. Coal plants are more expensive to build than fuel oil power plants, but they are cheaper to operate.

If more than one power plant is owned, the amount of power generated can be divided between or among the plants in any fashion desired — as long as no plant generates more than one million kilowatts. The only requirement is that enough power be generated to meet demand. Demand is determined by the number of customers. In a normal year in the program each customer consumes one thousand kilowatts.

The kilowatt demand is displayed for the user. If more than one plant is being operated, the display will subtract the power you chose to generate from previous plants and displays only the remaining kilowatts needed to meet demand.

If the amount of power generated falls short of the customers' demand then a brown-out occurs. This does not terminate the game, but damage is done to the generators and the company is charged an amount of money for repairs. The more the generated amount of electricity is below the demand, the more it will cost to repair the damage.

If the kilowatt demand outstrips the generating capacity of the company, then the generators go "poof!" and the game is terminated.

As power plants get older, they become increasingly inefficient. Each year the president of the company has to decide whether or not to remodel the plant(s). To aid in making the decision, a print out of cost for each plant, the percent of increase in efficiency for each plant, and total cost is shown. If the decision is to remodel, all plants are remodeled at the same time. The longer one waits to remodel, the more it will cost. The cost of remodeling is in part based on the increase of efficiency. Remodeling does not increase each plant's capacity. It makes each plant more efficient; it will use less fuel to make a kilowatt of electricity.

Very little in the program is allowed to remain constant. The population does not remain stable throughout the game. Each year, there is growth in the number of customers. Usually, the number of customers will increase at a rate that will require the building of a second power plant before the ten year term is completed.

Not only is the population not stable, but neither is the economy. An inflationary spiral is built into the price of all goods and services except one — the rate the company charges per kilowatt of electricity. As the program advances through each year, fuel, power plants, damage charges, etc. become more expensive.

Although the price per kilowatt the company gets for electricity does not automatically increase with inflation, a provision is made for applying for a rate increase each year. Rate increases are not allowed in two consecutive years. By waiting two years a rate increase request will be accepted if it is not too large. If it exceeds the upper limit defined by the program (which is tied to the Consumer Price Index), the request will be denied. If three or more years elapse between rate increases not only will the increase be accepted if it falls within the program's guide lines, but if it is too large, the program will yield a lesser compromise rate increase. Since an increase is not allowed every year, the upper limit for an allowable increase exceeds the Consumer Price Index.

Each year there is a 50 percent chance of some unforeseen event. These are not always bad or disastrous. Demand may rise a little faster or slower than usual, the company may be charged to repair storm damage to equipment, customers may be lost, additional customers may want to be hooked into the system (it costs the company a one-time fee to add these customers so the option of refusing service to them is given). If there are any costs involved in any of the above events, they will be influenced by inflation also.

Not all decisions become effective immediately. When ordering coal and fuel oil there is a lead time of one year. The fuel does not become available for use until the following year. Enough fuel must be on hand to supply the current year's needs. The same lead time of one year applies to rate increases also.

Two reports are shown. The first report, at the beginning of each year, shows the cash reserve, coal and fuel oil amounts, number of customers and number of plants owned. The second report, at the end of each year, displays the gross income, the maintenance costs for the year (maintenance costs are influenced by number of plants being operated, number of customers being served and the Consumer Price Index), percent of generating capacity that is being used, the inflation rate for the past year and the consumer price index.

Program Notes

THE ELECTRIC COMPANY is written in Radio Shack Level II Basic and requires 11K of memory. The instructions and scenario (not shown in the sample run) are located in lines 90-500.

The left hand brackets in lines 1820 and 1830 are the line printer's representation of an exponential arrow. If the Basic you have access to does not support exponentiation, substitute R*PI*SQR(PI). This yields the same result.

The "#" after some of the variables such as CR# indicates double precision. This prevents the output from appearing in scientific notation if the value of the variable exceeds six digits.

The strings defined in lines 10 and 20 are used in conjunction with the PRINTUSING statements to format the print-out of data by inserting dollar signs, commas and decimal points, or restricting numbers to two decimal places. If the PRINTUSING command is not available to you, the lines may be rewritten with only PRINT statements.

For those of you who wish to modify the degree of difficulty of the program, line 540 defines the initial values of the variables that are to be managed. □

Ralph G. White, 529 South Vermont, Columbus, KS 66725

```
10 D$="$$###,###,###.##":X$="   ###,###,###":C$="$$###.##"
20 E$="$$#####,###.##"
30 CLS:PRINTCHR$(23):PRINT:PRINT:PRINT
40 A$="+---++---++---++---++---++---++---++---+"
50 PRINTA$:PRINT:PRINT"          ELECTRIC COMPANY":PRINT:PRINT:PRINTA$
60 FORJ=1TO1200:NEXT
70 CLS:INPUT"DO YOU NEED INSTRUCTIONS (YES/NO) ";Z$
80 IFZ$="NO"GOTO540
90 CLS:PRINT"YOU HAVE BEEN HIRED AS PRESIDENT OF AN ELECTRIC COMPANY. "
100 PRINT"YOU WILL BE PRESIDENT FOR 10 YEARS BEFORE BEING NAMED TO THE"
110 PRINT"BOARD OF DIRECTORS--ASSUMING A SUCCESSFUL TERM AS PRESIDENT. "
120 PRINT"     YOU WILL START WITH ONE MILLION DOLLARS IN CASH, 1 POWER"
130 PRINT"PLANT AND STOCK PILES OF COAL AND FUEL OIL--THE TWO FUELS YOU"
140 PRINT"MAY USE TO POWER A GENERATING PLANT.  YOU MAY CHOOSE EITHER"
150 PRINT"FUEL YOU WISH TO POWER THE PLANT YOU START WITH AND EACH"
160 PRINT"SUCCEEDING PLANT YOU BUILD THEREAFTER.  ALL PLANTS HAVE A"
170 PRINT"GENERATING CAPACITY OF 1 MILLION KILOWATTS.  AS THE PLANTS"
180 PRINT"BECOME OLDER, THEY BECOME INCREASINGLY INEFFICIENT.  THAT IS,"
190 PRINT"THEY REQUIRE MORE FUEL TO PRODUCE EACH KILOWATT OF ELECTRICITY. "
200 PRINT"THEY CAN BE BROUGHT BACK TO PEAK EFFICIENCY BY REMODELING THEM. "
210 PRINT"THE LONGER YOU WAIT TO REMODEL, THE MORE IT WILL COST.  AND IF"
220 PRINT"MORE THAN ONE PLANT IS OWNED, ALL PLANTS MUST BE REMODELED AT"
230 PRINT"THE SAME TIME. "
240 INPUT"PRESS 'ENTER' TO CONTINUE INSTRUCTIONS ";Z$
250 CLS:PRINT"YOU HAVE TWO FUELS FROM WHICH TO CHOOSE: COAL AND FUEL OIL. "
260 PRINT"COAL TENDS TO YIELD MORE KILOWATTS PER DOLLAR OF FUEL; HOWEVER, "
270 PRINT"COAL PLANTS COST ABOUT 50% MORE TO BUILD THAN FUEL OIL PLANTS. "
280 PRINT"     MANAGEMENT OF FUEL SUPPLIES IS IMPORTANT BECAUSE THERE IS"
290 PRINT"A LEAD TIME OF 1 YEAR BEFORE IT ARRIVES.  COAL ORDERED IN YEAR"
300 PRINT"4 IS NOT AVAILABLE FOR USE UNTIL YEAR 5, SO A CLOSE EYE MUST BE"
310 PRINT"KEPT ON FUEL RESERVES. "
320 PRINT"     INFLATION AFFECTS THE COST OF ALMOST EVERYTHING YOU"
330 PRINT"ENCOUNTER, FROM FUEL COSTS TO DAMAGE CAUSED BY DISASTER OR"
340 PRINT"JUDGEMENT ERROR.  IF YOU FIND THAT A RATE INCREASE IS REQUIRED"
350 PRINT"TO MAINTAIN A PROFITABLE BUSINESS AND KEEP THE STOCK HOLDERS"
360 PRINT"HAPPY, YOU MAY APPLY FOR A RATE INCREASE EACH YEAR.  ALL YOU DO"
370 PRINT"IS SPECIFY WHAT YOU WANT THE NEW RATE TO BE (THE INITIAL RATE"
380 PRINT"IS .05 PER KILOWATT). "
390 PRINT:INPUT"PRESS 'ENTER' TO CONTINUE INSTRUCTIONS ";Z$
400 CLS:PRINT"     TEMPER YOUR RATE REQUESTS WITH MODERATION.  GREED WILL"
410 PRINT"GET YOU NOWHERE.  RATE INCREASES CAN BE REJECTED AS WELL AS"
420 PRINT"ACCEPTED.  BY WAITING LONGER BETWEEN INCREASE REQUESTS, YOU"
430 PRINT"CAN ASK FOR A LARGER INCREASE AND HAVE IT ACCEPTED.  ALSO BY"
440 PRINT"WAITING LONGER BETWEEN REQUESTS, A COMPROMISE RATE INCREAS MAY"
450 PRINT"BE GRANTED--WHERE AS FREQUENT REQUESTS MAY RESULT IN A DENIED"
460 PRINT"INCREASE.  AS WITH FUEL ORDERS, THERE IS A LEAD TIME OF 1 YEAR"
470 PRINT"FOR RATE HIKES.  A RATE INCREASE GRANTED IN YEAR 6 DOES NOT"
480 PRINT"TAKE AFFECT UNTIL YEAR 7. "
490 PRINT:PRINT"     NOW LIKE ALL GOOD EXECUTIVES, YOU MUST GET TO WORK. "
500 PRINT"              G O O D    L U C K ! ! !"
510 DEFDBLB-D,K-L
520 DEFDBLB-D,K-L
530 PRINT:PRINT:INPUT"PRESS 'ENTER' TO BEGIN ";Z$
540 YR=0:PI=1:GF=1:CS=700:CC=20:CF=25:CL=1600:FO=2500:CR#=1000000:R=.05:RT=
550 CLS.PRINT"WHAT FUEL DO YOU WANT TO POWER THE GENERATORS OF YOUR PLANT?"
560 INPUT"1=COAL   2=FUEL OIL";F(1)
570 IFF(1)<1ORF(1)>2GOTO550
580 EF(1)=1
590 YR=YR+1
600 I=0
610 E=(100+RND(10))/100
620 I=I+1:EF(I)=EF(I)*E:IFI<GFGOTO610
630 CLS:PRINT"   ***** FINANCES FOR BEGINNING OF YEAR ";YR;" *****":PRINT
640 PRINT"GENERATING PLANTS OWNED              ";:PRINTUSINGX$;GF
650 PRINT"NUMBER OF CUSTOMERS                  ";:PRINTUSINGX$;CS
660 PRINT"TONS OF COAL IN STOCK                ";:PRINTUSINGX$;CL
670 PRINT"BARRELS OF FUEL OIL IN STOCK         ";:PRINTUSINGX$;FO
680 PRINT"CASH RESERVE                         ";:PRINTUSINGD$;CR#
690 PRINT:INPUT"PRESS 'ENTER' TO BEGIN THE YEAR'S TRANSACTIONS";Z$
700 IFCR#>0GOTO780
710 CLS.PRINT"YOU HAVE BUNGLED THE JOB.  THE COMPANY IS BROKE. "
720 PRINT"YOU HAVE BEEN SENT TO A SIBERIAN SALT MINE TO COOL YOUR HEELS. "
730 PRINT GOTO2220
740 PRINT"YOU HAVE MISMANAGED YOUR FUEL INVENTORY.  THAT IS TOO BAD, FOR"
750 PRINT"YOU WERE DOING FINE IN OTHER AREAS OF MANAGEMENT   BUT THE"
760 PRINT"BOARD OF DIRECTORS WERE UNSYMPATHETIC, YOU WERE CANNED. "
770 PRINT:GOTO2220
780 IN=(100+RND(12))/100:PI=PI*IN:PG=(100+RND(12))/100
790 MN=((GF*5000)+(CS*10))*PI.CR#=CR#-MN
800 CS=INT(CS*PG):KD#=CS*1000:GC=1000000*GF
810 CC=INT(2000*PI)/100:SC=INT(100*PI*(RND(30)+20)/10)/100
820 OC=INT(2500*PI)/100 SO=INT(100*PI*(RND(10)+10)/10)/100
830 EV=RND(10):IFEV>5GOTO1000
840 ONEVGOTO850  ,920  ,940  ,960  ,980
850 CA=RND(100) HU=INT((500+RND(500)+(1/CA)*RND(500))*PI)
860 CLS PRINTCA;" PEOPLE HAVE REQUESTED TO BE ADDED TO THE CUSTOMER LIST. "
870 PRINT"IT WILL COST $";HU;" TO HOOK THE CUSTOMERS TO YOUR SYSTEM. "
880 PRINT"DO YOU WISH TO SELL POWER TO THEM?":INPUT"1=YES   2=NO";DC
890 IFDC<1ORDC>2GOTO860
900 IFDC=2GOTO1000
910 CS=CS+CA:GOTO1000
920 CA=RND(100):CLS:PRINTCA;" PEOPLE HAVE DROPPED AS CUSTOMERS"
930 CS=CS-CA:GOTO990
940 SD=(RND(1000)+1000)*PI:PRINT"A STORM CAUSED $";SD;" DAMAGE. "
950 CR#=CR#-SD:GOTO990
960 PRINT"PEOPLE PROMISE TO CONSERVE.  DEMAND WILL NOT RISE AS SHARPLY"
970 PRINT"THIS YEAR.":KD#=INT(KD#*.9):GOTO990
980 PRINT"DEMAND WILL RISE FASTER THAN USUAL ":KD#=INT(KD#*1.1)
990 FORJ=1TO2000:NEXT
1000 KG=0:NC=0:NF=0:I=0
1010 I=I+1
1020 CLS:PRINTUSINGX$;(KD#-KG);
1030 PRINT" KILOWATTS ARE NEEDED.  YOU HAVE ";GF;" PLANTS. "
1040 PRINT"HOW MANY KILOWATTS FROM PLANT ";I;:INPUT" ";K
1050 IFK<=1000000GOTO1070
1060 PRINT"ONE PLANT CAN NOT PRODUCE THAT MUCH.":GOTO1040
1070 KG=KG+K
1080 IFF(I)=2GOTO1120
1090 F=INT((K/600*EF(I))+.5):NC=NC+F:IFNC<CLGOTO1110
1100 NC=NC-F:KG=KG-K:PRINT"YOU DO NOT HAVE ENOUGH COAL.":GOTO740
1110 PRINT"PLANT ";I;" NEEDS ";F;" TONS OF COAL.":GOTO1150
1120 F=INT((K/400*EF(I))+.5):NF=NF+F:IFNF<FOGOTO1140
1130 NF=NF-F:KG=KG-K:PRINT"YOU DID NOT HAVE ENOUGH FUEL OIL.":GOTO740
1140 PRINT"PLANT ";I;" NEEDS ";F;" BARRELS OF OIL. "
1150 FORJ=1TO1500:NEXT
1160 IFI<GFGOTO1010
1170 IFKD#>GCGOTO1200
1180 IFKD#>KGGOTO1240
1190 GOTO1280
1200 CLS:PRINT"ALL GENERATORS WERE OVERLOADED!  DEMAND WAS GREATER THAN"
1210 PRINT"CAPACITY.  A MAJOR BLACKOUT OCCURRED.  SEVERE DAMAGE HAS BEEN"
1220 PRINT"CAUSED TO THE GENERATORS AND YOUR CAREER.  YOU HAVE BEEN SENT"
1230 PRINT"TO ANTARCTICA TO MANAGE AN ICE CREAM PARLOR.":GOTO2220
1240 DM=(RND(1000)+2000)*PI*KD#/KG:CR=CR-DM
1250 CLS:PRINT"YOU DID NOT GENERATE ENOUGH TO MEET DEMANDS. "
1260 PRINT"$";DM;" DOLLARS WORTH OF DAMAGE HAS BEEN CAUSED TO THE"
1270 PRINT"EQUIPMENT.  FORTUNATELY THIS WAS NOT A FATAL MISTAKE. "
1280 PRINT"FUEL REQUIREMENTS"
1290 PRINTNC;" TONS OF COAL":PRINTNF;" BARRELS OF FUEL OIL";PRINT
1300 PRINT"COST PER TON OR BARREL";TAB(30);"SHIPPING";TAB(45);"TOTAL COST"
1310 PC=CC+SC:PO=OC+SO
1320 PRINT"COAL";TAB(10);:PRINTUSINGC$;CC;
1330 PRINTTAB(30);:PRINTUSINGC$;SC;
1340 PRINTTAB(50);:PRINTUSINGC$;PC
1350 PRINT"FUEL OIL";TAB(10);:PRINTUSINGC$;OC;
1360 PRINTTAB(30);:PRINTUSINGC$;SO;
1370 PRINTTAB(50);:PRINTUSINGC$;PO
1380 INPUT"HOW MANY TONS OF COAL DO YOU WISH TO BUY ";CP
1390 INPUT"HOW MANY BARRELS OF FUEL OIL DO YOU WISH TO BUY ";OP
1400 CB=CP*PC:OB=OP*PO
1410 CLS:PRINT"FUEL COSTS";TAB(20);"COAL";TAB(35);"FUEL OIL";TAB(50);"TOTA
1420 PRINTTAB(15);:PRINTUSINGE$;CB;:PRINTTAB(30);:PRINTUSINGE$;OB;
     :PRINTTAB(45);:PRINTUSINGE$;(CB+OB)
1430 ER#=KG*RT:CR#=CR#-CB-OB+ER#:CL=CL+CP-NC:FO=FO+OP-NF
1440 FORJ=1TO2000:NEXT
1450 CLS:PRINT"REMODELING DOES NOT INCREASE THE GENERATING CAPACITY OF A"
1460 PRINT"PLANT.  IT WILL RESTORE ALL PLANTS TO PEAK EFFICIENCY.  ALL"
1470 PRINT"PLANTS MUST BE DONE AT THE SAME TIME.":PRINT
1480 PRINT"PLANT";TAB(20);"COST";TAB(40);"INCREASE IN EFFICIENCY"
1490 RM=0
1500 FORI=1TOGF
1510 RC=5000*EF(I)*PI:SV=((EF(I)-1)/EF(I))*100
1520 RM=RM+RC
1530 PRINTI;TAB(15);:PRINTUSINGE$;RC;:PRINTTAB(50);SV;" %"
1540 NEXT
1550 PRINT"TOTAL COST     ";:PRINTUSINGE$;RM
1560 PRINT:PRINT"DO YOU WISH TO REMODEL?":INPUT"1=YES   2=NO";DC
1570 IFDC<1ORDC>2GOTO1560
1580 IFDC=2GOTO1610
1590 CR#=CR#-RM
1600 FORI=1TOGF:EF(I)=1:NEXT
1610 CM=(RND(5000)+45000)*PI:OM=(RND(3000)+30000)*PI
1620 CLS:PRINT"A COAL GENERATING PLANT WILL COST     ";:PRINTUSINGD$;CM
1630 PRINT"A FUEL OIL GENERATING PLANT WILL COST ";:PRINTUSINGD$;OM
1640 PRINT:PRINT"DO YOU WISH TO BUILD A NEW PLANT?"
1650 PRINT"1=COAL   2=FUEL OIL   3=NEITHER"
1660 INPUT"WHAT IS YOUR DECISION ";DC
1670 IFDC<1ORDC>3GOTO1640
1680 IFDC>1GOTO1700
1690 CR#=CR#-CM:GF=GF+1:F(GF)=1:EF(GF)=1:GOTO1720
1700 IFDC>2GOTO1720
1710 CR#=CR#-OM:GF=GF+1:F(GF)=2:EF(GF)=1
1720 CLS:PRINT"DO YOU WISH TO APPLY FOR A RATE INCREASE?"
1730 INPUT"1=YES   2=NO ";DC
1740 IFDC<1ORDC>2GOTO1720
1750 IFDC=1GOTO1770
1760 GOTO1920
1770 PRINT PRINT"IT HAS BEEN ";TR;" YEARS SINCE YOU HAD A RAISE "
1780 PRINT"CUSTOMERS ARE PRESENTLY PAYING ";RT;"  PER KILOWATT. "
```

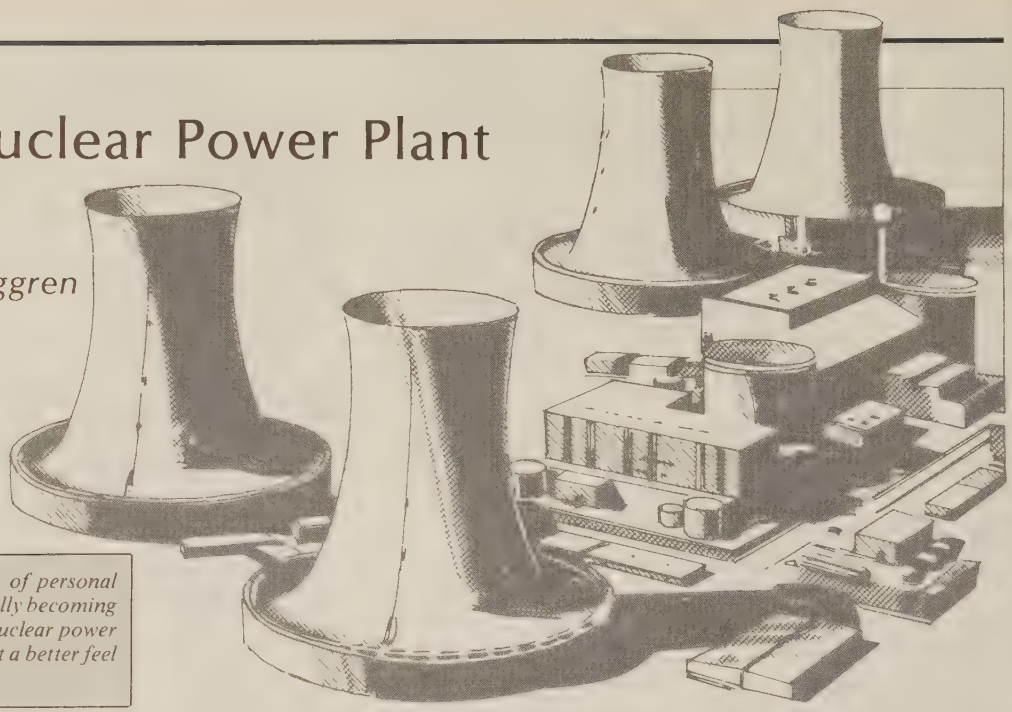
```
1790 PRINT"HOW MUCH DO YOU WISH TO CHARGE PER KILOWATT?"
1800 INPUT"ENTER REQUEST AS A DECIMAL (.07 OR .075 ETC.)";PR
1810 IFTR<2GOTO1890
1820 IFPR>R*PI1.5ANDTR<3GOTO1890
1830 IFPR<R*PI1.5GOTO1870
1840 RT=RT+((PR-RT)*.8):TR=0
1850 PRINT"YOU ARE ALLOWED TO INCREASE YOUR RATE TO ";RT;" PER KILOWATT."
1860 GOTO1910
1870 PRINT"YOUR RATE INCREASE HAS BEEN STUDIED AND ACCEPTED AS REQUESTED"
1880 RT=PR:TR=0:GOTO1910
1890 PRINT"THE RATE INCREASE YOU REQUESTED HAS BEEN DENIED."
1900 PRINT"NO INCREASE IS ALLOWED "
1910 FORJ=1TO1500:NEXT
1920 TR=TR+1
1930 CLS:PRINT"      ***** END OF YEAR ";YR;" REPORT *****":PRINT
1940 PRINT"INCOME FROM SALE OF POWER          ";:PRINTUSINGD$;ER#
1950 PRINT"POWER PLANTS OPERATED              ";:PRINTUSINGX$;GF
1960 PRINT"MAINTAINENCE COSTS                 ";:PRINTUSINGD$;MN
1970 PRINT"YOU ARE GENERATING AT  ";(KG/GC)*100;"% OF CAPACITY"
1980 PRINT"THE INFLATION RATE FOR THE YEAR WAS      ";(IN*100)-100;"%"
1990 PRINT"THE CONSUMER PRICE INDEX NOW STANDS AT   ";PI
2000 PRINT:PRINT:INPUT"PRESS 'ENTER' TO BEGIN THE NEXT YEAR";Z$
2010 IFYR<10GOTO590
2020 CLS:PRINT"     ***** ENDING FINANCIAL REPORT *****":PRINT
2030 PRINT"GENERATING PLANTS OWNED            ";:PRINTUSINGX$;GF
2040 PRINT"NUMBER OF CUSTOMERS                ";:PRINTUSINGX$;CS
2050 PRINT"TONS OF COAL IN STOCK              ";:PRINTUSINGX$;CL
2060 PRINT"BARRELS OF FUEL OIL IN STOCK       ";:PRINTUSINGX$;FO
2070 PRINT"CASH RESERVE                       ";:PRINTUSINGD$;CR#
2080 PRINT:PRINT"     ************************************"
2090 PRINT:PRINT:INPUT"PRESS 'ENTER' FOR COMMENTS ";Z$:CLS
2100 PRINT"CONGRATULATIONS!  YOU LASTED THE 10 YEARS."
2110 AS=((GF-1)*40000+CR+CL*25+FO*25)*PI
2120 IFAS>1000000*PI*.95GOTO2160
2130 PRINT"YOU SURVIVED AND THAT WAS ALL.  YOUR ASSETS DID NOT GROW AS"
2140 PRINT"FAST AS INFLATION.  INSTEAD OF BEING A MEMBER OF THE BOARD OF"
2150 PRINT"DIRECTORS, YOU HAVE BEEN GIVEN A POSITION AS A JANITOR.":GOTO2220
2160 IFAS>1000000*PI*1.1GOTO2190
2170 PRINT"YOU KEPT PACE WITH INFLATION.  YOU HAVE BEEN GIVEN THE POSITION"
2180 PRINT"AS MEMBER OF THE BOARD OF DIRECTORS.":GOTO2220
2190 PRINT"NOT ONLY ARE YOU APPOINTED A MEMBER OF THE BOARD OF DIRECTORS"
2200 PRINT"BUT YOU HAVE BEEN MADE CHAIRMAN OF THE BOARD AND AS A BONUS"
2210 PRINT"YOU HAVE BEEN GIVEN A YEAR'S SUPPLY OF SMELLY CIGARS."
2220 END
```

```
WHAT FUEL DO YOU WANT TO POWER THE GENERATORS OF YOUR PLANT? 1

+--++--++--++--++--++--++--++--+

***** FINANCES FOR BEGINNING OF YEAR  1  *****
GENERATING PLANTS OWNED                      1
NUMBER OF CUSTOMERS                        700
TONS OF COAL IN STOCK                    1,600
BARRELS OF FUEL OIL IN STOCK             2,500
CASH RESERVE                        $1,000,000.00
PEOPLE PROMISE TO CONSERVE.  DEMAND WILL NOT RISE AS SHARPLY
THIS YEAR.

     686,699 KILOWATTS ARE NEEDED.  YOU HAVE  1  PLANTS.
HOW MANY KILOWATTS FROM PLANT  1  680000
PLANT  1  NEEDS  1167  TONS OF COAL.

YOU DID NOT GENERATE ENOUGH TO MEET DEMANDS.
$ 2268.77  DOLLARS WORTH OF DAMAGE HAS BEEN CAUSED TO THE
EQUIPMENT.  FORTUNATELY THIS WAS NOT A FATAL MISTAKE.
FUEL REQUIREMENTS
 1167  TONS OF COAL
 0  BARRELS OF FUEL OIL
COST PER TON OR BARREL        SHIPPING       TOTAL COST
COAL       $22.20              $3.77             $25.97
FUEL OIL   $27.75              $1.77             $29.52
HOW MANY TONS OF COAL DO YOU WISH TO BUY?  1500
HOW MANY BARRELS OF FUEL OIL DO YOU WISH TO BUY?  200

 COSTS          COAL          FUEL OIL       TOTAL
              $38,955.00      $5,904.00     $44,859.00

REMODELING DOES NOT INCREASE THE GENERATING CAPACITY OF A
PLANT.  IT WILL RESTORE ALL PLANTS TO PEAK EFFICIENCY.  ALL
PLANTS MUST BE DONE AT THE SAME TIME.
PLANT              COST                INCREASE IN EFFICIENCY
 1              $5,716.50                        2.91262  %
TOTAL COST      $5,716.50
DO YOU WISH TO REMODEL?  1=YES  2=NO  2

 COAL GENERATING PLANT WILL COST          $53,174.60
A FUEL OIL GENERATING PLANT WILL COST      $35,217.00
DO YOU WISH TO BUILD A NEW PLANT?  1=COAL  2=FUEL OIL  3=NEITHER  2

DO YOU WISH TO APPLY FOR A RATE INCREASE?  1=YES  2=NO  1

IT HAS BEEN  0  YEARS SINCE YOU HAD A RAISE.
CUSTOMERS ARE PRESENTLY PAYING  .05  PER KILOWATT.
HOW MUCH DO YOU WISH TO CHARGE PER KILOWATT?  .06
THE RATE INCREASE YOU REQUESTED HAS BEEN DENIED.
NO INCREASE IS ALLOWED.


     ***** END OF YEAR  1  REPORT *****
INCOME FROM SALE OF POWER              $34,000.00
MAINTAINENCE COSTS                     $13,320.00
YOU ARE GENERATING AT  68 % OF CAPACITY
THE INFLATION RATE FOR THE YEAR WAS      11 %
THE CONSUMER PRICE INDEX NOW STANDS AT   1.11
+--++--++--++--++--++--++--++--+

   ***** FINANCES FOR BEGINNING OF YEAR  2  *****
GENERATING PLANTS OWNED                      2
NUMBER OF CUSTOMERS                        763
TONS OF COAL IN STOCK                    1,933
BARRELS OF FUEL OIL IN STOCK             2,700
CASH RESERVE                          $940,604.03

     808,000 KILOWATTS ARE NEEDED.  YOU HAVE  2  PLANTS.
HOW MANY KILOWATTS FROM PLANT  1  750000
PLANT  1  NEEDS  1391  TONS OF COAL.

      58,000 KILOWATTS ARE NEEDED.  YOU HAVE  2  PLANTS.
HOW MANY KILOWATTS FROM PLANT  2  58000
PLANT  2  NEEDS  151  BARRELS OF OIL.
FUEL REQUIREMENTS
 1391  TONS OF COAL
 151  BARRELS OF FUEL OIL
COST PER TON OR BARREL        SHIPPING       TOTAL COST
COAL       $23.75              $3.08             $26.83
FUEL OIL   $29.69              $2.01             $31.70
HOW MANY TONS OF COAL DO YOU WISH TO BUY?  1500
HOW MANY BARRELS OF FUEL OIL DO YOU WISH TO BUY?  300

L COSTS          COAL          FUEL OIL       TOTAL
               $40,245.00      $9,510.00     $49,755.00

REMODELING DOES NOT INCREASE THE GENERATING CAPACITY OF A
PLANT.  IT WILL RESTORE ALL PLANTS TO PEAK EFFICIENCY.  ALL
PLANTS MUST BE DONE AT THE SAME TIME.
PLANT              COST                INCREASE IN EFFICIENCY
 1              $6,605.99                       10.1043  %
 2              $6,176.04                        3.84615 %
TOTAL COST      $12,782.00
DO YOU WISH TO REMODEL?  1=YES  2=NO  2

 GENERATING PLANT WILL COST          $58,685.40
A FUEL OIL GENERATING PLANT WILL COST      $39,078.90
DO YOU WISH TO BUILD A NEW PLANT?  1=COAL  2=FUEL OIL  3=NEITHER  2

DO YOU WISH TO APPLY FOR A RATE INCREASE?  1=YES  2=NO  2

     ***** END OF YEAR  2  REPORT *****
INCOME FROM SALE OF POWER              $40,400.00
MAINTAINENCE COSTS                     $20,939.20
YOU ARE GENERATING AT  40.4 % OF CAPACITY
THE INFLATION RATE FOR THE YEAR WAS      7.00001 %
THE CONSUMER PRICE INDEX NOW STANDS AT   1.1877


+--++--++--++--++--++--++--++--+

   ***** FINANCES FOR BEGINNING OF YEAR  3  *****
GENERATING PLANTS OWNED                      3
NUMBER OF CUSTOMERS                        808
TONS OF COAL IN STOCK                    2,042
BARRELS OF FUEL OIL IN STOCK             2,849
CASH RESERVE                          $871,230.98
PEOPLE PROMISE TO CONSERVE.  DEMAND WILL NOT RISE AS SHARPLY
THIS YEAR.

     813,599 KILOWATTS ARE NEEDED.  YOU HAVE  3  PLANTS.
HOW MANY KILOWATTS FROM PLANT  1  800000
PLANT  1  NEEDS  1557  TONS OF COAL.

      13,599 KILOWATTS ARE NEEDED.  YOU HAVE  3  PLANTS.
HOW MANY KILOWATTS FROM PLANT  2  13599
PLANT  2  NEEDS  36  BARRELS OF OIL.

           0 KILOWATTS ARE NEEDED.  YOU HAVE  3  PLANTS.
HOW MANY KILOWATTS FROM PLANT  3  0
PLANT  3  NEEDS  0  BARRELS OF OIL.
FUEL REQUIREMENTS
 1557  TONS OF COAL
 36  BARRELS OF FUEL OIL
COST PER TON OR BARREL        SHIPPING       TOTAL COST
```

```
COAL        $26.36                $3.16                $29.52
FUEL OIL    $32.95                $2.50                $35.45
HOW MANY TONS OF COAL DO YOU WISH TO BUY?  1200
HOW MANY BARRELS OF FUEL OIL DO YOU WISH TO BUY?  300

UEL COSTS           COAL           FUEL OIL       TOTAL
                 $35,424.00      $10,635.00     $46,059.00

REMODELING DOES NOT INCREASE THE GENERATING CAPACITY OF A
PLANT.  IT WILL RESTORE ALL PLANTS TO PEAK EFFICIENCY.  ALL
PLANTS MUST BE DONE AT THE SAME TIME.
PLANT                COST                  INCREASE IN EFFICIENCY
 1                $7,699.28                        14.385  %
 2                $7,061.07                        6.64674  %
 3                $6,987.24                        5.66037  %
TOTAL COST       $21,747.60
DO YOU WISH TO REMODEL?  1=YES  2=NO  1

 COAL GENERATING PLANT WILL COST          $62,720.40
A FUEL OIL GENERATING PLANT WILL COST      $40,719.80
DO YOU WISH TO BUILD A NEW PLANT?  1=COAL  2=FUEL OIL  3=NEITHER  3

DO YOU WISH TO APPLY FOR A RATE INCREASE?  1=YES  2=NO  1
IT HAS BEEN  2  YEARS SINCE YOU HAD A RAISE.
CUSTOMERS ARE PRESENTLY PAYING  .05  PER KILOWATT.
HOW MUCH DO YOU WISH TO CHARGE PER KILOWATT?  .06
YOUR RATE INCREASE HAS BEEN STUDIED AND ACCEPTED AS REQUESTED


     ***** END OF YEAR  3  REPORT *****
INCOME FROM SALE OF POWER              $40,679.95
MAINTAINENCE COSTS                     $30,427.50
YOU ARE GENERATING AT  27.12 % OF CAPACITY
THE INFLATION RATE FOR THE YEAR WAS      11 %
THE CONSUMER PRICE INDEX NOW STANDS AT  1.31835


+--++--++--++--++--++--++--++--+

 ***** FINANCES FOR BEGINNING OF YEAR  4  *****
GENERATING PLANTS OWNED                      3
NUMBER OF CUSTOMERS                        904
TONS OF COAL IN STOCK                    1,685
BARRELS OF FUEL OIL IN STOCK             3,113
CASH RESERVE                          $813,676.90


     976,000 KILOWATTS ARE NEEDED.  YOU HAVE  3  PLANTS.
HOW MANY KILOWATTS FROM PLANT  1  900000
PLANT  1  NEEDS  1650  TONS OF COAL.

      76,000 KILOWATTS ARE NEEDED.  YOU HAVE  3  PLANTS.
HOW MANY KILOWATTS FROM PLANT  2  50000
PLANT  2  NEEDS  136  BARRELS OF OIL.

      26,000 KILOWATTS ARE NEEDED.  YOU HAVE  3  PLANTS.
HOW MANY KILOWATTS FROM PLANT  3  26000
PLANT  3  NEEDS  68  BARRELS OF OIL.
FUEL REQUIREMENTS
 1650  TONS OF COAL
 204  BARRELS OF FUEL OIL
COST PER TON OR BARREL        SHIPPING       TOTAL COST
COAL        $27.42              $5.89             $33.31
FUEL OIL    $34.27              $2.60             $36.87
HOW MANY TONS OF COAL DO YOU WISH TO BUY?  1000
HOW MANY BARRELS OF FUEL OIL DO YOU WISH TO BUY?  500

FUEL COSTS          COAL           FUEL OIL       TOTAL
                 $33,310.00      $18,435.00     $51,745.00

REMODELING DOES NOT INCREASE THE GENERATING CAPACITY OF A
PLANT.  IT WILL RESTORE ALL PLANTS TO PEAK EFFICIENCY.  ALL
PLANTS MUST BE DONE AT THE SAME TIME.
PLANT                COST                  INCREASE IN EFFICIENCY
 1                $7,540.94                        9.09091  %
 2                $7,472.39                        8.25688  %
 3                $7,129.62                        3.84615  %
TOTAL COST       $22,143.00
DO YOU WISH TO REMODEL?  1=YES  2=NO  2

AL GENERATING PLANT WILL COST          $68,067.30
A FUEL OIL GENERATING PLANT WILL COST      $41,763.10
DO YOU WISH TO BUILD A NEW PLANT?  1=COAL  2=FUEL OIL  3=NEITHER  3

DO YOU WISH TO APPLY FOR A RATE INCREASE?  1=YES  2=NO  2

     ***** END OF YEAR  4  REPORT *****
INCOME FROM SALE OF POWER              $58,560.00
MAINTAINENCE COSTS                     $32,960.80
YOU ARE GENERATING AT  32.5333 % OF CAPACITY
THE INFLATION RATE FOR THE YEAR WAS      4 %
THE CONSUMER PRICE INDEX NOW STANDS AT  1.37108

--++--++--++--++--++--++--++--+

   ***** FINANCES FOR BEGINNING OF YEAR  5  *****
GENERATING PLANTS OWNED                      3
NUMBER OF CUSTOMERS                        976
TONS OF COAL IN STOCK                    1,035
BARRELS OF FUEL OIL IN STOCK             3,409
CASH RESERVE                          $787,531.12


    1,054,000 KILOWATTS ARE NEEDED.  YOU HAVE  3  PLANTS.
HOW MANY KILOWATTS FROM PLANT  1  900000
YOU DO NOT HAVE ENOUGH COAL.
YOU HAVE MISMANAGED YOUR FUEL INVENTORY.  THAT IS TOO BAD, FOR
YOU WERE DOING FINE IN OTHER AREAS OF MANAGEMENT.  BUT THE
BOARD OF DIRECTORS WERE UNSYMPATHETIC, YOU WERE CANNED.
```

Apple Nuclear Power Plant

Stephen R. Berggren



The political implications of personal computers are only gradually becoming clear. By trying to run a nuclear power plant, for example, you get a better feel for the risks involved . . .

The safety, usefulness and desirability of nuclear power plants is a subject that has recently been a very "hot" topic of conversation. It seems that almost everyone has an opinion about what should be done about nuclear power plants. It's too bad that only a very few people have actually been at the controls of one of the big generating plants to get a firsthand feel for how it works.

That's all I wanted to do; to run my own nuclear power plant. Unfortunately, I did not have several hundred million dollars or a license from the Nuclear Regulatory Commission. Neither did I want to wait the ten years it would take to build a power plant. I was therefore forced to use what I did have, my Apple II computer. This wasn't such a bad alternative. After all, on a computer the worst thing that could result from my mismanagement would be a software crash. I would be completely safe from explosions, radiation and irate protesters.

My efforts at a safe study of nuclear power have resulted in a program that simulates the performance of a pressurized water nuclear power plant. The program is designed to be educational, as well as challenging and fun. It starts with a simple description of the workings of a plant I designed. This is followed by an animated diagram of the plant and its operation in color graphics. Then instructions are provided on how to operate the controls. Finally the plant starts operation with a daily status report appearing on the screen. The keyboard is used to adjust the controls and then move to the next day. Within the program, a series of equations simulates the response of the reactor and power system and presents the next day's status. Warnings are provided when the plant is not operating properly or when any damage is done. This report is a much simplified version of the computerized status reporting used in most nuclear power plants.

As with a real power plant, the object is to produce as much electrical power as possible without injuring anyone or damaging the equipment. This plant can produce up to two megawatts of power before overloading, but the average power output depends on how close to this maximum the plant is operated and how long the plant is down for maintenance. When the plant's nuclear fuel is exhausted (after about 120 days) the program evaluates the operator's performance based on the average power output and the damage sustained by the plant. Of course, if the plant's operation should result in a meltdown, the operator should consider his performance rather poor.

A Disastrous Sample Run

In the sample program run, you can see that at day one the plant was in cold shutdown and everything was at 25 (degrees centigrade). Of course, the control rods were fully inserted and all coolant flow was stopped. I started the plant by pulling out the control rods. As the reactor heat went up, primary and secondary coolant flow was started to cool the reactor and run the generator turbine. By day 15, the plant's temperatures had settled down and the plant was running well, although not very close to its maximum power output. That's the trick; keeping the power output high and the temperatures in the operating ranges.

At day 19, a small coolant leak developed, but I decided not to shut down just to fix it. However, by day 23 things had really turned sour, and I had to try for a maintenance shutdown. With all coolant systems on full and the control rods in, the reactor cooled quickly and then entered the automatic maintenance shutdown mode. It took 32 days to repair the damage but then the plant was ready to start again. But I made a mistake. I tried to heat the reactor too quickly and went beyond the safe temperature range. Then, I tried to compensate by increasing the primary coolant flow and got into more trouble. Before I could get the emergency coolant on, the reactor went out of control and disaster struck. Too bad, but I deserved it. At least no one was hurt.

The program is written in a very straightforward manner and only the keyboard input and operating algorithms should be at all hard to understand. Lines 955 to 985 demonstrate the keyboard input. The key is entered as A$ and filtered in line 965 to allow only (space), (return) and 0 thru 9 to pass through. The (space) will skip to the next variable while the (return) will allow the program to continue to the next day's report. A$ is then concatenated to B$ whose VAL becomes the new input variable. The FOR statement in line 955 limits the input to four keystrokes. The program is almost entirely crashproof. *(Famous last words.—Ed.)*

If you are really interested in the operating algorithm, lines 1280 to 1395 simulate heat flow equations that have been simplified slightly. Don't feel too bad if you don't understand exactly how they work. They do work. The REM statements

Stephen R. Berggren, 104 Ridgeway Ave. Louisville,KY 40207

at the end of the program give the prefixes and suffixes used to create the variables. They also describe what each section is supposed to do.

Program Details

The program is written in Applesoft, Apple's floating point Basic, and fits in a 16K memory. Translation should be very easy with some exceptions. The diagram routines use color graphics. If your system does not have graphics, delete lines 6000 to 7060. You must also fix lines 220 and 222, since calling a deleted subroutine is an easy way to crash. The program also makes extensive use of logic evaluations within expressions. For example, (A<1) equals one if true, and zero if false. This is a quick and easy way to avoid IF statements. If your system does not have this capability, convert each of these expressions to IF statements. Many of the variables have percent signs following them. This is Applesoft's way of saying 'integer variable.' I used them to keep fractions out of the numbers printed in the status report. If your system can easily control the number of decimal places printed, forget the percent signs.

Before I could get the emergency coolant on, the reactor went out of control and disaster struck. Too bad, but I deserved it.

In Applesoft, the PEEK in line 910 returns the vertical position of the cursor. If you can't find your cursor, you will have to think up another way to input the control variable. Finally, the instructions and the status report are made to fit a 40 by 24 character screen. These can be easily modified to fill a wider screen.

I have some final words to engineers, nuclear technicians and other qualified readers. No, I have never seen a real power plant that was designed like this. Yes, I know it takes much less than a day for a reactor to respond to changes in coolant flow and control rod position. Besides, who ever heard of a reactor with only one emergency cooling system, and that one with a limited supply of coolant? And, blasphemy of blasphemies, no SCRAM mechanism? I am sure your list of discrepancies is far more complete. What I have tried to do is to incorporate characteristics and responses of a simplified and idealized nuclear power plant into a computer simulation game. Several concessions to accuracy were made in order to create a simulation that would provide realistic responses to simple inputs and make an interesting and instructive game. □

```
10  HOME
20  PRINT  SPC( 8)"APPLE NUCLEAR POWER PLANT"
30  PRINT  SPC( 9)"BY STEPHEN R. BERGGREN
40  PRINT
50  PRINT "THIS PROGRAM SIMULATES THE OPERATION OF"
60  PRINT "A NUCLEAR POWER REACTOR.  THE OBJECT"
70  PRINT "IS TO OPERATE THE PLANT AT A MAXIMUM"
80  PRINT "AVERAGE POWER OUTPUT WITHOUT CAUSING"
90  PRINT "A REACTOR MELTDOWN."
100  PRINT
110  PRINT "THE CONTROL RODS ADJUST THE AMOUNT OF"
120  PRINT "HEAT PRODUCED BY THE REACTOR.  PRIMARY"
130  PRINT "COOLANT TRANSFERS THIS HEAT TO THE HEAT"
140  PRINT "EXCHANGER.  SECONDARY COOLANT TRANSFERS"
150  PRINT "HEAT FROM THE HEAT EXCHANGER TO THE"
160  PRINT "TURBINE, WHERE POWER IS PRODUCED, AND"
170  PRINT "FINALLY TO THE COOLING TOWER.  THE"
180  PRINT "EMERGENCY COOLANT IS USED TO HELP SHUT"
190  PRINT "DOWN THE REACTOR WHEN OTHER SYSTEMS"
200  PRINT "FAIL.  UNLIKE THE OTHER COOLANTS, "
210  PRINT "EMERGENCY COOLANT IS NOT RECYCLED."
220  PRINT : INPUT "ENTER 'D' TO SEE REACTOR DIAGRAM    ENTER 'I' FOR WORKIN
G INSTRUCTIONS      ENTER 'S' TO START OPERATION     ";A$
221  IF A$ = "D" THEN  GOSUB 6000: GOTO 220
222  IF A$ = "S" THEN 390
225  TEXT : HOME
230  PRINT "THE CONTROLS ARE OPERATED BY TYPING IN"
235  PRINT "THE DESIRED CONTROL ROD SETTING AND"
240  PRINT "FLOW RATES.(USE VALUES FROM 0 TO 100)"
245  PRINT "IF NO ENTRY IS MADE, THE VALUES WILL"
250  PRINT "NOT CHANGE.  USE THE SPACE BAR TO STEP"
255  PRINT "TO THE DIFFERENT FUNCTIONS.  WHEN THE"
260  PRINT "DESIRED ENTRIES HAVE BEEN MADE, USE THE"
265  PRINT "'RETURN' KEY TO ADVANCE TO THE NEXT DAY."
270  PRINT "THE REACTOR CAN BE OPERATED UNTIL A"
275  PRINT "MELTDOWN OCCURS OR THE REACTOR FUEL IS"
280  PRINT "EXHAUSTED.  THE FUEL WILL LAST FOR"
285  PRINT "ABOUT 100 TO 150 DAYS.  WHEN THE FUEL"
290  PRINT "IS EXHAUSTED, YOUR PERFORMANCE WILL BE"
295  PRINT "EVALUATED."
298  PRINT : INPUT "   (PRESS RETURN TO CONTINUE)";A$: HOME
300  PRINT : PRINT "IF YOU WANT TO REPAIR DAMAGE OR REPLACE"
305  PRINT "COOLANT, BRING THE REACTOR TEMPERATURE"
310  PRINT "DOWN BELOW 100 AND SHUT OFF THE COOLANT"
315  PRINT "FLOWS.  THIS WILL CAUSE AN AUTOMATIC"
320  PRINT "MAINTENANCE SHUTDOWN AND ALL COOLANT"
325  PRINT "WILL BE REPLENISHED AND REPAIRS MADE."
330  PRINT "THE GREATER THE DAMAGE, THE LONGER THE"
335  PRINT "REPAIRS WILL TAKE."
340  PRINT
350  PRINT "     WARNING: THIS POWER PLANT HAS"
360  PRINT "     NO AUTOMATIC SAFETY DEVICES!!"
370  PRINT
380  GOTO 220
390  REM  INITIATE
400  GOSUB 2000
410 RH = 0
420 RL = 0
430 DAY% = 0
440 TT = 0
450 DMGE% = 0
455 A% = 0:A1% = 0:A2% = 0
460  REM WRITE REPORT
470  TEXT : HOME
475 DAY% = DAY% + 1
480  PRINT  SPC( 7)"APPLE NUCLEAR POWER PLANT"
490  PRINT  SPC( 8)"STATUS REPORT - DAY ";DAY%
500  PRINT
510  PRINT "WARNINGS:"
520  IF RT% > 800 THEN  PRINT " REACTOR OVERHEATED":RD% = RD% + 1 + (RT%
     > 850) + (RT% > 900) + 2 * (RT% > 950):PD% = PD% + 1:ED% = ED% + 1
     + (RT% > 850)
530  IF XT% > 500 THEN  PRINT " HEAT EXCHANGER OVERHEATED":XD% = XD% + 1
     + (XT% > 600):PD% = PD% + 1:SD% = SD% + 1
540  IF GO% > 2000 THEN  PRINT " TURBINE OVERLOADED":TD% = TD% + 1 + (GO%
     > 2500  ):SD% = SD% + 1
550  IF CT% > 300 THEN  PRINT " COOLING TOWER OVERHEATED":SD% = SD% + 1
560  IF GO% < 1000 THEN  PRINT " POWER OUTPUT LOW"
570  IF EU% < 200 THEN  PRINT " EMERGENCY COOLANT LOW"
580  IF PU% < 100 THEN  PRINT " PRIMARY COOLANT LOW":PD% = PD% + 1
590  IF SU% < 100 THEN  PRINT " SECONDARY COOLANT LOW":SD% = SD% + 1
600  PRINT
610  PRINT "DAMAGE:"
620  IF RD% > 3 THEN  PRINT " REACTOR CORE DAMAGED"
630  IF PD% > 4 THEN  PRINT " PRIMARY COOLANT LEAK - ";PD%;"/DAY":PU% =
     (PU% - P
D%) * ((PU% - PD%) > 0)
640  IF SD% > 4 THEN  PRINT " SECONDARY COOLANT LEAK - ";SD%;"/DAY":SU% =
```

```
     (SU% - SD%) * ((SU% - SD%) > 0)
650  IF ED% > 2 THEN  PRINT " EMERGENCY COOLANT LEAK - ";2 * ED%;"/DAY":
     EU% = (E U% - 2 * ED%) * ((EU% - 2 * ED%) > 0)
660  IF PB% THEN  PRINT " PRIMARY COOLANT PUMP FAILURE - ";10 * PD% *
     (PD% < 10)  + 100 * (PD% >  = 10);"%"
670  IF SB% THEN  PRINT " SECONDARY COOLANT PUMP FAILURE - ";10 * SD% *
     (SD% < 1  0) + 100 * (SD% >  = 10);"%"
680  IF XB% THEN  PRINT " HEAT EXCHANGER FAILURE"
690  IF GB% THEN  PRINT " TURBINE FAILURE"
700  PRINT
710  IF RD% > 5 THEN  PRINT "      MELTDOWN!  MELTDOWN!  MELTDOWN!":
     GOTO 3000
720  PRINT "INDICATORS:"
730  PRINT " REACTOR TEMP. (MAX 800) ";RT%
740  PRINT " HEAT EXCHANGER TEMP. (MAX 500) ";XT%
750  PRINT " COOLING TOWER TEMP. (MAX 300) ";CT%
760  PRINT " POWER OUTPUT (MAX 2000KW) ";GO%;"KW"
765 KW% = TT / DAY%
770  PRINT " AVERAGE POWER OUTPUT  ";KW%;"KW/DAY
800  PRINT " CONTROL RODS- ";A%
810  PRINT " COOLANTS"
820  PRINT "  EMERGENCY  LEVEL- ";EU%;"  FLOW- ";EF%
830  PRINT "  PRIMARY    LEVEL- ";PU%;"  FLOW- "PF%
840  PRINT "  SECONDARY  LEVEL- ";SU%;"  FLOW- "SF%;
850  IF (100 - RL) < 5 THEN  PRINT : PRINT : PRINT "REACTOR FUEL
     EXHAUSTED": GOTO 4000
900  REM GET NEW CONTROL VALUES
910 P =  PEEK (37)
920  VTAB (P - 3)
930  HTAB (20)
950 A2% = A1%:A1% = A%
955 B$ = "": FOR I = 1 TO 4
960  GET A$:Z =  ASC (A$)
965  IF (Z <  > 13 AND Z <  > 32) AND (Z > 57 OR Z < 48) THEN 960
970 B$ = B$ + A$: IF Z = 13 THEN 1170
975  IF Z = 32 THEN 990
980 A% =  VAL (B$):A% = A% + (100 - A%) * (A% > 100)
985  PRINT A$;: NEXT I
990  VTAB (P - 1)
1000  HTAB (35)
1005 B$ = "": FOR I = 1 TO 4
1010  GET A$:Z =  ASC (A$)
1015  IF (Z <  > 13 AND Z <  > 32) AND (Z > 57 OR Z < 48) THEN 1010
1020 B$ = B$ + A$: IF Z = 13 THEN 1170
1025  IF Z = 32 THEN 1050
1030 EF% =  VAL (B$):EF% = EF% + (100 - EF%) * (EF% > 100)
1035  IF EF% > EU% THEN EF% = EU%
1040  PRINT A$;: NEXT I
1050  VTAB (P)
1060  HTAB (35)
1065 B$ = "": FOR I = 1 TO 4
1070  GET A$:Z =  ASC (A$)
1075  IF (Z <  > 13 AND Z <  > 32) AND (Z > 57 OR Z < 48) THEN 1070
1080 B$ = B$ + A$: IF Z = 13 THEN 1170
1085  IF Z = 32 THEN 1110
1090 PF% =  VAL (B$):PF% = PF% + (100 - PF%) * (PF% > 100)
1100  PRINT A$;: NEXT I
1110  VTAB (P + 1)
1120  HTAB (35)
1125 B$ = "": FOR I = 1 TO 4
1130  GET A$:Z =  ASC (A$)
1135  IF (Z <  > 13 AND Z <  > 32) AND (Z > 57 OR Z < 48) THEN 1130
1140 B$ = B$ + A$: IF Z = 13 THEN 1170
1145  IF Z = 32 THEN 1165
1150 SF% =  VAL (B$):SF% = SF% + (100 - SF%) * (SF% > 100)
1160  PRINT A$;: NEXT I
1165  HTAB (1): VTAB (P - 3): CALL  - 958: GOTO 800
1170  IF PF% = 0 AND SF% = 0 AND RH < 1 AND RT% < 100 AND A% = 0 THEN
      GOSUB 200 0: HTAB (1): VTAB (24): CALL  - 922: PRINT "
      MAINTENANCE SHUTDOWN - ";MD%;" D
1180  IF EF% > EU% THEN EF% = EU%
1200  REM DAMAGE ASSESSMENT AND OPERATION CALCULATIONS
1205 EU% = EU% - EF% - 2 * ED% * (ED% > 3)
1210 PD% = PD% + (PF% > 90) * ( RND (20) > .95)
1220 SD% = SD% + (SF% > 90) * ( RND (20) > .92)
1230 PB% = PD% > 5
1240 SB% = SD% > 5
1250  IF PF% > (100 - PD% * 10) AND PB% THEN PF% = (100 - PD% * 10) *
      (100 - PD%  * 10 > 0)
1260  IF SF% > (100 - SD% * 10) AND SB% THEN SF% = (100 - SD% * 10) *
      (100 - SD%  * 10 > 0)
1270 RL = RL + RH / 50
1280 RH = (A% * 30 + A1% * 60 + A2% * 10) / 2500 * (100 - RL)
1300 PH = PF% * (100 * (PU% > 100) + PU% * (PU% <  = 100)) / 350
1310 EH = EF% / 200 * (RT% - 25)
1320 RT% = RT% + RH - EH - PH - 5 * (RT% > 25)
1325 RT% = 25 + (RT% - 25) * (RT% > 25)
1330 XT% = ((RT% - 25) * PF% + (CT% - 25) * SF%) / (PF% + SF% + 1) + 25
```

```
1340  IF XB% THEN XT% = RT% * .8 + 5
1350 SH = SF% * (100 * (PU% > 100) + PU% * (PU% <  = 100)) / 350 *
     (XT% - CT%)
1360  IF XB% THEN SH = SH * .2
1370 GO% = SH / XT% * (XT% - CT%) * 2 / 3
1375  IF GO% > 2600 THEN GO% = 2600
1380 GO% = GO% * (GO% > 0) * (GB% = 0)
1390 CT% = 25 + ((XT% - 25) * (SH - GO%) / (SH + 1) * .75)
1395 CT% = 25 * (CT% <  = 25) + CT% * (CT% > 25)
1400  IF XB% < 1 THEN XB% = (XD% > 2) * ( RND (4) > .9)
1410  IF GB% < 1 THEN GB% = (GD% > 4) * ( RND (4) > .9)
1420 TT = TT + GO%
1430  GOTO 470
2000  REM MAINTENANCE REPAIR SUBROUTINE
2010 EV% = 300
2020 PV% = 120
2030 SV% = 120
2040 RT% = 25
2050 XT% = 25
2060 CT% = 25
2070 DMGE% = DMGE% + 2 * RD% + ED% + PD% + XD% + SD% + GD%
2080 MD% = 5 + 3 * (10 * (RD% > 3) + (ED% > 3) + (PD% > 3) + (SD% > 3)
     + 2 * PB%  + 2 * SB% + 3 * XB% + 3 * GB%):DAY% = DAY% + MD%
2090 RD% = 0
2100 ED% = 0
2110 PD% = 0
2120 XD% = 0
2130 SD% = 0
2140 GD% = 0
2150 PB% = 0
2160 SB% = 0
2170 XB% = 0
2180 GB% = 0
2190 EF% = 0:PF% = 0:SF% = 0
2195 GO% = 0
2200  RETURN
3000  REM  MELTDOWN ENDING
3010  PRINT
3020  IF RD% > 6 THEN 3100
3030  PRINT "THE REACTOR CORE HAS BEEN DISTROYED BY"
3040  PRINT "UNCONTROLLED THERMAL RUNAWAY.  HOWEVER,"
3050  PRINT "THE CONTAINMENT BUILDING HAS NOT YET"
3060  PRINT "RUPTURED."
3070  PRINT
3080  PRINT "INITIATE YOUR EVACUATION PLAN."
3090  GOTO 5000
3100  PRINT "THE REACTOR CORE HAS MELTED DOWN AND"
3110  PRINT "PRODUCED A STEAM EXPLOSION.  THE"
3120  PRINT "CONTAINMENT BUILDING HAS RUPTURED."
3130  PRINT "LETHAL RADIOACTIVE GASES AND DEBRIS"
3140  PRINT "HAVE ESCAPED."
3150  PRINT
3160  PRINT "INITIATE YOUR EVACUATION AND RADIATION"
3170  PRINT "CLEANUP PLANS AND GET MEDICAL"
3180  PRINT "ASSISTANCE."
3190  GOTO 5000
4000  REM  EVALUATION OF GAME RESULTS
4010  PRINT
4020  PRINT "OVER A PERIOD OF ";DAY%;" DAYS, YOU HAVE"
4030  PRINT "PRODUCED AN AVERAGE POWER OUTPUT OF"
4040  PRINT KW%;" KILOWATTS PER DAY."
4050 AKW% = 1 + (KW% > 1000) + (KW% > 1200) + (KW% > 1500) + (KW% > 1800)
4060  PRINT
4070  PRINT "YOUR AVERAGE POWER PRODUCTION RATE IS"
4080  ON AKW% GOTO 4090,4100,4110,4120,4140
4090  PRINT "HORRIBLE! FIND A LESS DEMANDING JOB.": GOTO 4200
4100  PRINT "WAY BELOW YOUR AREA'S POWER NEEDS.": GOTO 4200
4110  PRINT "ADEQUATE. YOU COULD DO BETTER.": GOTO 4200
4120  PRINT "EXCELLENT! POWER COSTS IN YOUR AREA"
4130  PRINT "WILL NOT BE INCREASED.": GOTO 4200
4140  PRINT "NEAR THE MAXIMUM!  POWER COSTS IN YOUR"
4150  PRINT "AREA WILL DROP SIGNIFICANTLY."
4200  REM DAMAGE EVALUATION
4210  PRINT
4215  GOSUB 2000
4220 D% = 1 + (DMGE% > 10) + (DMGE% > 20) + (DMGE% > 30)
4230  PRINT "THE EQUIPMENT DAMAGE SUSTAINED DURING"
4240  PRINT "THIS PERIOD WAS ";
4250  ON D% GOTO 4260,4270,4280,4290
4260  PRINT "VERY LIGHT.": GOTO 5000
4270  PRINT "MODERATE.": GOTO 5000
4280  PRINT "HEAVY.": GOTO 5000
4290  PRINT "SEVERE."
5000  REM END
5010  PRINT
5020  PRINT "WOULD YOU LIKE TO TRY AGAIN? (Y OR N)";
5030  INPUT A$
5040  IF A$ = "" THEN 5030
5050  IF A$ = "Y" THEN  GOSUB 2000: GOTO 390
5060  HOME
5070  END
5100 B$ = ""
```

```
5105  FOR I = 1 TO 4
5110  GET A$
5130  IF  ASC (A$) <  > 32 THEN 5140
5134  IF B$ = "" THEN  RETURN
5136 OUT% =  VAL (B$)
5138  RETURN
5140  IF  ASC (A$) = 13 THEN  POP : GOTO 1170
5150 B$ = B$ + A$
5160  PRINT A$;
5170  NEXT I
5180  RETURN
6000  GR : COLOR= 15: FOR I = 0 TO 39: HLIN 0,39 AT I: NEXT I
6005 DE = 6000
6010  HOME : VTAB 23
6020  PRINT "  THIS IS THE REACTOR VESSEL"
6030  RESTORE
6035  COLOR= 1
6040  FOR I = 1 TO 29
6050  READ X1,X2,Y
6060  HLIN X1,X2 AT Y
6070  NEXT I
6080  DATA 7,9,6,15,17,6,6,18,7,5,19,8,5,7,9,17,19,9,5,6,10,18,19,10,5,6,
      11,18,19,11,5,6,12,18,19,12,5,6,13,18,19,13,5,6,14
6090  DATA 18,19,14,5,6,15,18,19,15,5,6,16,18,19,16,5,6,17,18,19,17,5,6,
      18,18,19 ,18,5,7,19,17,19,19,6,18,20,7,17,21,8,16,22
6100  COLOR= 0
6110  HLIN 13,14 AT 6
6120  COLOR= 12
6130  HLIN 14,15 AT 5
6140  HLIN 14,16 AT 4
6150  HLIN 15,16 AT 3
6160  FOR I = 1 TO DE: NEXT I
6170  VTAB 23
6180  PRINT "  THIS IS THE REACTOR CORE     "
6190  COLOR= 8
6200  FOR I = 9 TO 15
6210  VLIN 11,17 AT I
6220  NEXT I
6230  FOR I = 1 TO DE: NEXT I
6240  VTAB 23
6250  PRINT "  THESE ARE THE CONTROL RODS  "
6260  COLOR= 13
6270  VLIN 2,17 AT 11
6280  VLIN 2,17 AT 13
6290  FOR I = 1 TO DE: NEXT I
6300  VTAB 23
6310  PRINT " THE EMERGENCY COOLANT CAN COOL THE"
6320  PRINT " REACTOR IN AN EMERGENCY."
6330  COLOR= 2
6340  FOR I = 1 TO 36
6350  READ Y,X
6360  PLOT X,Y
6370  FOR J = 1 TO 200: NEXT J
6380  NEXT I
6390  DATA 4,2,4,4,5,2,5,3,5,4,6,2,6,3,6,4,7,3,8,3,9,3,10,3,11,3,12,3,
      12,4,12,5, 12,6,12,7,12,8,12,9
6400  DATA 13,9,14,9,15,9,16,9,16,8,16,7,16,6,16,5,16,4,16,3,17,3,18,3,
      19,3,20,3 ,22,3,24,3
6410  FOR I = 1 TO DE: NEXT I
6420  HOME : VTAB 23
6430  PRINT " THE PRIMARY COOLANT CARRIES HEAT FROM"
6440  PRINT " THE REACTOR CORE TO THE HEAT EXCHANGER"
6450  FOR I = 1 TO 52
6460  READ Y,X
6470  PLOT X,Y
6480  FOR J = 1 TO 200: NEXT J
6490  NEXT I
6500  FOR I = 1 TO DE: NEXT I
6510  DATA 4,25,4,27,5,25,5,26,5,27,6,25,6,26,6,27,7,26,8,26,9,26,10,26,
      11,26,12 ,26,12,25,12,24,12,23,12,22,12,21,12,20,12,19
6515  DATA 12,18,12,17,12,16,12,15,13,15,14,15,15,15,16,15,16,16,16,
      17,16,18
6520  DATA  16,19,16,20,16,21,16,22,16,23,16,24,16,25,16,26,16,27,16,28,
      16,29,16 ,30,15,30,14,30,13,30,13,30,12,30,12,29,12,28,12,27
6540  HOME : VTAB 23
6550  PRINT " THIS IS THE HEAT EXCHANGER"
6560  COLOR= 5
6570  HLIN 28,34 AT 10
6580  VLIN 10,18 AT 34
6590  HLIN 28,34 AT 18
6600  VLIN 10,18 AT 28
6605  COLOR= 2: PLOT 28,12: PLOT 28,16: COLOR= 5
6610  FOR I = 1 TO DE: NEXT I
6620  VTAB 23
6630  PRINT " THIS IS THE GENERATOR TURBINE"
6640  HLIN 5,18 AT 30
6650  VLIN 30,36 AT 18
6660  HLIN 5,18 AT 36
6670  VLIN 30,36 AT 5
```

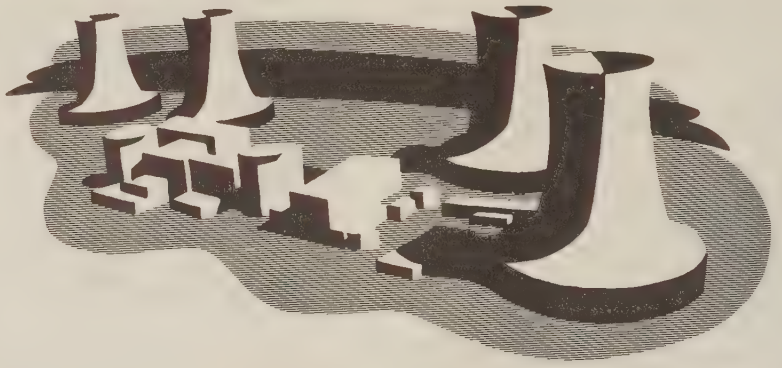
```
6680  COLOR= 0
6690  HLIN 2,17 AT 33
6700  FOR I = 7 TO 15 STEP 2
6710  PLOT I,34: PLOT I + 1,32
6720  NEXT I
6730  FOR I = 1 TO DE: NEXT I
6740  VTAB 23
6750  PRINT " THIS IS THE COOLING TOWER           "
6760  COLOR= 5
6770  VLIN 23,25 AT 24
6780  VLIN 23,25 AT 36
6790  VLIN 25,26 AT 25
6800  VLIN 25,26 AT 35
6810  VLIN 26,28 AT 26
6820  VLIN 26,28 AT 34
6830  VLIN 28,36 AT 27
6840  VLIN 28,36 AT 33
6850  PLOT 34,36
6860  PLOT 26,36
6870  HLIN 25,35 AT 38
6880  FOR I = 1 TO DE: NEXT I
6890  VTAB 23
6900  PRINT " THE SECONDARY COOLANT CARRIES HEAT"
6910  PRINT " FROM THE HEAT EXCHANGER TO THE "
6920  PRINT " TURBINE AND THEN TO THE COOLING TOWER"
6925  COLOR= 2
6930  FOR I = 1 TO 123
6940  READ Y,X
6950  PLOT X,Y
6960  FOR J = 1 TO 200: NEXT J
6970  NEXT I
6980  DATA 4,35,4,37,5,35,5,36,5,37,6,35,6,36,6,37,7,36,8,36,9,36,10,36,
      11,36,12,36,12,35,12,34,12,33,12,32,13,32,14,32,15,32,16,32
6990  DATA 16,33,16,34,16,35,16,36,17,36,18,36,19,36,20,36,20,35,20,34,
      20,33,20,32,20,31,20,30,20,29,20,28,20,27,20,26,20,25,20,24,20,23,
      20,22,20,21
7000  DATA 21,21,22,21,23,21,24,21,25,21,26,21,27,21,27,20,27,19,27,18,
      27,17,27,16,27,15,27,14,27,13,27,12,27,11,27,10,27,9,27,8,27,7
7010  DATA 28,7,29,7,30,7,31,7,32,7,34,10,32,13,34,16,35,16,35,17,35,18,
      35,19,35,20,35,21,35,22,35,23,35,24,35,25,35,26,35,27,35,28
7020  DATA 34,28,34,29,34,30,35,30,35,31,35,32,34,32,34,33,34,34,34,35,
      34,36,34,37,34,38,33,38,32,38,31,38,30,38,29,38,28,38,27,38,26,38,
      25,38,24,38,23,38
7030  DATA 22,38,21,38,20,38,19,38,18,38,17,38,16,38,15,38,14,38,13,38,
      12,38,12,37
7040  HOME : VTAB 23
7060  RETURN
9000  REM VARIABLE PREFIXES
9010  REM A-CONTROL RODS, C-COOLING TOWER, E-EMERGENCY COOLANT,
      G-TURBINE, P-PRIMARY COOLANT, R-REACTOR, S-SECONDARY COOLANT,
      X-HEAT EXCHANGER
9020  REM VARIABLE SUFFIXES
9030  REM B-BROKEN, D-DAMAGE, F-FLOW RATE, H-HEAT FLOW, L-LIFE, O-OUTPUT,
      T-TEMPERATURE, V-VOLUME
9040  REM OTHER VARIABLES TOT-TOTAL POWER OUTPUT, KW-AVERAGE POWER
      OUTPUT, DAY-DAY OF OPERATION, DMGE-TOTAL EQUIPMENT DAMAGE
9050  REM PROGRAM DISCRIPTION BY LINE NUMBER
9060  REM 10-220 INTRODUCTION
9070  REM 225-380 INSTRUCTIONS
9080  REM 390-455 VARIABLE INITIATION
9090  REM  460-850 WRITE REPORT AND ASSESS DAMAGE
9100  REM  900-1165 INPUT NEW CONTROL VARIABLES
9110  REM 1170 MAINTENANCE SHUTDOWN EVALUATION
9120  REM 1200-1260 PUMP FAILURE ASSESSMENT
9130  REM 1270-1430 PLANT OPERATING ALGORITHMS
9140  REM 2000-2200 MAINTENANCE SHUTDOWN SUBROUTINE
9150  REM 3000-3190 MELTDOWN ENDING
9160  REM 4000-4290 EVALUATION OF GAME RESULTS
9170  REM 5000-5070 END
9180  REM 6000-7060 PLANT DIAGRAM SUBROUTINE
9190  REM 9000-9190 REMARKS
9200  REM  APPLE NUCLEAR POWER PLANT
9210  REM  BY STEPHEN R BERGGREN

RUN

        APPLE NUCLEAR POWER PLANT
         BY STEPHEN R. BERGGREN

THIS PROGRAM SIMULATES THE OPERATION OF
A NUCLEAR POWER REACTOR.  THE OBJECT
IS TO OPERATE THE PLANT AT A MAXIMUM
AVERAGE POWER OUTPUT WITHOUT CAUSING
A REACTOR MELTDOWN.

THE CONTROL RODS ADJUST THE AMOUNT OF
HEAT PRODUCED BY THE REACTOR.  PRIMARY
COOLANT TRANSFERS THIS HEAT TO THE HEAT
EXCHANGER.  SECONDARY COOLANT TRANSFERS
HEAT FROM THE HEAT EXCHANGER TO THE
TURBINE, WHERE POWER IS PRODUCED, AND
FINALLY TO THE COOLING TOWER.
```

```
THE  EMERGENCY COOLANT IS USED TO HELP SHUT
DOWN THE REACTOR WHEN OTHER SYSTEMS
FAIL.  UNLIKE THE OTHER COOLANTS,
EMERGENCY COOLANT IS NOT RECYCLED.

ENTER 'D' TO SEE REACTOR DIAGRAM      ENTER 'I' FOR WORKING INSTRUCTIONS
ENTER 'S' TO START OPERATION
THE CONTROLS ARE OPERATED BY TYPING IN
THE DESIRED CONTROL ROD SETTING AND
FLOW RATES.(USE VALUES FROM 0 TO 100)
IF NO ENTRY IS MADE, THE VALUES WILL
NOT CHANGE.  USE THE SPACE BAR TO STEP
TO THE DIFFERENT FUNCTIONS.  WHEN THE
DESIRED ENTRIES HAVE BEEN MADE, USE THE
'RETURN' KEY TO ADVANCE TO THE NEXT DAY.
THE REACTOR CAN BE OPERATED UNTIL A
MELTDOWN OCCURS OR THE REACTOR FUEL IS
EXHAUSTED.  THE FUEL WILL LAST FOR
ABOUT 100 TO 150 DAYS.  WHEN THE FUEL
IS EXHAUSTED, YOUR PERFORMANCE WILL BE
EVALUATED.

    (PRESS RETURN TO CONTINUE)

IF YOU WANT TO REPAIR DAMAGE OR REPLACE
COOLANT, BRING THE REACTOR TEMPERATURE
DOWN BELOW 100 AND SHUT OFF THE COOLANT
FLOWS.  THIS WILL CAUSE AN AUTOMATIC
MAINTENANCE SHUTDOWN AND ALL COOLANT
WILL BE REPLENISHED AND REPAIRS MADE.
THE GREATER THE DAMAGE, THE LONGER THE
REPAIRS WILL TAKE.

      WARNING: THIS POWER PLANT HAS
      NO AUTOMATIC SAFETY DEVICES!!


ENTER 'D' TO SEE REACTOR DIAGRAM   ENTER 'I' FOR WORKING INSTRUCTIONS
ENTER 'S' TO START OPERATION         S
        APPLE NUCLEAR POWER PLANT
         STATUS REPORT - DAY 1

WARNINGS:
 POWER OUTPUT LOW

DAMAGE:

INDICATORS:
 REACTOR TEMP. (MAX 800) 25
 HEAT EXCHANGER TEMP. (MAX 500) 25
 COOLING TOWER TEMP. (MAX 300) 25
 POWER OUTPUT (MAX 2000KW) 0KW
 AVERAGE POWER OUTPUT  0KW/DAY
 CONTROL RODS- 0
 COOLANTS
  EMERGENCY  LEVEL- 300  FLOW- 0
  PRIMARY    LEVEL- 120  FLOW- 0
  SECONDARY  LEVEL- 120  FLOW- 01
        STATUS REPORT - DAY 2

WARNINGS:
 POWER OUTPUT LOW

DAMAGE:

INDICATORS:
 REACTOR TEMP. (MAX 800) 26
 HEAT EXCHANGER TEMP. (MAX 500) 25
 COOLING TOWER TEMP. (MAX 300) 25
 POWER OUTPUT (MAX 2000KW) 0KW
 AVERAGE POWER OUTPUT  0KW/DAY
 CONTROL RODS- 1
 COOLANTS
  EMERGENCY  LEVEL- 300  FLOW- 0
  PRIMARY    LEVEL- 120  FLOW- 0
  SECONDARY  LEVEL- 120  FLOW- 0100



        APPLE NUCLEAR POWER PLANT
         STATUS REPORT - DAY 19

WARNINGS:

DAMAGE:
 SECONDARY COOLANT LEAK - 5/DAY

INDICATORS:
 REACTOR TEMP. (MAX 800) 778
 HEAT EXCHANGER TEMP. (MAX 500) 465
```

```
 COOLING TOWER TEMP. (MAX 300) 254
 POWER OUTPUT (MAX 2000KW) 1858KW
 AVERAGE POWER OUTPUT  1478KW/DAY
 CONTROL RODS- 9
 COOLANTS
  EMERGENCY  LEVEL- 300  FLOW- 0
  PRIMARY    LEVEL- 120  FLOW- 70
  SECONDARY  LEVEL- 105  FLOW- 100
        STATUS REPORT - DAY 20

       APPLE NUCLEAR POWER PLANT
        STATUS REPORT - DAY 23

WARNINGS:
 HEAT EXCHANGER OVERHEATED
 COOLING TOWER OVERHEATED
 POWER OUTPUT LOW
 SECONDARY COOLANT LOW

DAMAGE:
 SECONDARY COOLANT LEAK - 11/DAY
 SECONDARY COOLANT PUMP FAILURE - 100%

INDICATORS:
 REACTOR TEMP. (MAX 800) 783
 HEAT EXCHANGER TEMP. (MAX 500) 667
 COOLING TOWER TEMP. (MAX 300) 327
 POWER OUTPUT (MAX 2000KW) 790KW
 AVERAGE POWER OUTPUT  1487KW/DAY
 CONTROL RODS- 9
 COOLANTS
  EMERGENCY  LEVEL- 300  FLOW- 0
  PRIMARY    LEVEL- 120  FLOW- 70
  SECONDARY  LEVEL- 76  FLOW- 20

       APPLE NUCLEAR POWER PLANT
        STATUS REPORT - DAY 30

WARNINGS:
 POWER OUTPUT LOW
 EMERGENCY COOLANT LOW
 PRIMARY COOLANT LOW
 SECONDARY COOLANT LOW

DAMAGE:
 PRIMARY COOLANT LEAK - 7/DAY
 SECONDARY COOLANT LEAK - 19/DAY
 PRIMARY COOLANT PUMP FAILURE - 70%
 SECONDARY COOLANT PUMP FAILURE - 100%
 HEAT EXCHANGER FAILURE

INDICATORS:
 REACTOR TEMP. (MAX 800) 96
 HEAT EXCHANGER TEMP. (MAX 500) 81
 COOLING TOWER TEMP. (MAX 300) 25
 POWER OUTPUT (MAX 2000KW) 0KW
 AVERAGE POWER OUTPUT  1140KW/DAY
 CONTROL RODS- 0
 COOLANTS
  EMERGENCY  LEVEL- 0  FLOW- 0
  PRIMARY    LEVEL- 82  FLOW- 0
  SECONDARY  LEVEL- 0  FLOW- 0
       MAINTENANCE SHUTDOWN - 32 DAYS

       APPLE NUCLEAR POWER PLANT
        STATUS REPORT - DAY 63

       APPLE NUCLEAR POWER PLANT
        STATUS REPORT - DAY 69

WARNINGS:
 REACTOR OVERHEATED
 TURBINE OVERLOADED

DAMAGE:
 REACTOR CORE DAMAGED
 EMERGENCY COOLANT LEAK - 10/DAY

     MELTDOWN!  MELTDOWN!  MELTDOWN!

THE REACTOR CORE HAS MELTED DOWN AND
PRODUCED A STEAM EXPLOSION.  THE
CONTAINMENT BUILDING HAS RUPTURED.
LETHAL RADIOACTIVE GASES AND DEBRIS
HAVE ESCAPED.

INITIATE YOUR EVACUATION AND
RADIATION CLEANUP PLANS AND
GET MEDICAL ASSISTANCE.

WOULD YOU LIKE TO TRY AGAIN?
(Y OR N)?N
```

PET Nuclear Power Plant

Ivan Smith



Nuclear power plants are going to be with us for a long time, presenting challenges to their supporters and to their opponents. Many people have opinions about nuclear plants, but not many have a clear idea how they work.

A written description of something as complex as a nuclear power plant has a tendency to overwhelm people with layers of detail. There is no way to write a description of the interactions between the rate of heat generation in the reactor core, the rate of flow of primary and secondary coolants, the electrical power output of the turbine-generator, and the temperatures of the various components, and at the same time have a readable, comprehensible, reasonably accurate presentation which is interesting enough for most people to wade through. Here is an opportunity for microcomputers to show their stuff.

Simulation is a field in which microcomputers can make a major contribution toward a better-informed population. Computers certainly have a bright future in such applications as information retrieval and record-keeping, but their educational potential has not even begun to be developed, especially in the public schools. The ability of a computer to simulate a complex and fluid situation, and to involve a person directly in that situation, cannot be duplicated or even approximated by books, films, television, or by a human teacher—no matter how gifted.

For these reasons I was glad to find Stephen R. Berggren's nuclear power plant simulation program in the December 1980 issue of *Creative Computing*. This is a fine example of the sort of simulation which develops the unique educational capabilities of microcomputers. It tackles a topic which is vitally important to today's society, a topic which has been almost totally ignored in the public schools. Sure, the schools talk about radiation hazards from an environmental point of view, an important subject, but how many of this year's high school graduates will have even a foggy notion of how a nuclear power station *works*? For that matter, how many teachers—even science teachers—know?

Berggren's simulation makes it easy, even entertaining, to find out the basics of the inner workings of the heat cycle of a nuclear power plant.

The school in which I work has PET computers. Berggren's program is written for the Apple, so I undertook to adapt it for the PET. I found it took more effort to make this adaptation than I expected. Here is the result, so that this valuable simulation will be available to many who might want to use it, but do not have an Apple.

To maintain as much comparability as possible between this PET version and the original Apple version, the statements have not been renumbered, despite a certain higgledy-piggledy quality which has evolved. I have not tampered with the core of Berggren's program, the equations which describe the heat transfer characteristics of the plant, because they work very well as they are, and because I do not understand them well enough to attempt any changes.

The graphics, which begin at statement 6000, have been completely rewritten by my son, Hume Smith. Berggren's commentary on variables, which begins at statement 9000, is reproduced in Figure 1. His variables have been retained without change except as noted below.

New variables VE and VE$, statements 762-766, have been introduced to repre-

Figure 1.

```
9000  REM VARIABLE PREFIXES
9010  REM A-CONTROL RODS, C-COOLING TOWER, E-EMERGENCY COOLANT,
      G-TURBINE, P-PRIMARY COOLANT, R-REACTOR, S-SECONDARY COOLANT,
      X-HEAT EXCHANGER
9020  REM VARIABLE SUFFIXES
9030  REM B-BROKEN, D-DAMAGE, F-FLOW RATE, H-HEAT FLOW, L-LIFE, O-OUTPUT,
      T-TEMPERATURE, V-VOLUME
9040  REM OTHER VARIABLES TOT-TOTAL POWER OUTPUT, KW-AVERAGE POWER
      OUTPUT, DAY-DAY OF OPERATION, DMGE-TOTAL EQUIPMENT DAMAGE
9050  REM PROGRAM DISCRIPTION BY LINE NUMBER
9060  REM 10-220 INTRODUCTION
9070  REM 225-380 INSTRUCTIONS
9080  REM 390-455 VARIABLE INITIATION
9090  REM  460-850 WRITE REPORT AND ASSESS DAMAGE
9100  REM  900-1165 INPUT NEW CONTROL VARIABLES
9110  REM 1170 MAINTENANCE SHUTDOWN EVALUATION
9120  REM 1200-1260 PUMP FAILURE ASSESSMENT
9130  REM 1270-1430 PLANT OPERATING ALGORITHMS
9140  REM 2000-2200 MAINTENANCE SHUTDOWN SUBROUTINE
9150  REM 3000-3190 MELTDOWN ENDING
9160  REM 4000-4290 EVALUATION OF GAME RESULTS
9170  REM 5000-5070 END
9180  REM 6000-7060 PLANT DIAGRAM SUBROUTINE
9190  REM 9000-9190 REMARKS
9200  REM  APPLE NUCLEAR POWER PLANT
9210  REM  BY STEPHEN R BERGGREN
```

Ivan C. Smith, P.O. Box 226, Chester, N.S., Canada B0J 1J0.

sent the dollar value of the energy produced. This has been calculated at 5.5¢ per kWh, a value which can be changed in statement 762. VE$ is used to convert a number greater than 9999 to a format easy to read, with the digits grouped by threes. This grouping has traditionally been done by using a comma, but is done here using a space as is the style in metric usage. Users who may wish to use the traditional format with a comma between the thousands and ten-thousands digits, can alter the concatenation in statement 766 by inserting +","+ in place of +""+.

In the original program, commentary statement 9040 assigns TOT to represent the total energy output, but the operating variable appears to be TT, as in statements 765 and 1420. Variable BL$, used in the PET version in statements 40, 920, 960, 1000, 1120, etc, consists of 36 spaces followed by cursor home, and is used to clear one line on the screen.

In the original program, statements 1210 and 1220 are used to generate a random damage factor for the primary and secondary coolant loops. There is a 5% change of primary loop damage whenever the primary coolant flow exceeds 90%, and an 8% chance of secondary loop damage whenever the secondary coolant flow exceeds 90%. This damage shows up as a leakage rate, which is simply a count of the number of times the random factor occurs. The leakage accumulates in each coolant system, and a shutdown is forced when leakage reaches 5 in either system.

In running this program many times, I got the feeling that each run was much the same as every other run, and that the leakage factor was a bit too simple. All one had to do was keep the coolant flows below 90% and everything would go well almost indefinitely. I decided to alter the random leakage factors to a more interesting and a more realistic format, using three ideas: quality of workmanship, multi-step flow-dependent random damage, and age-dependent random damage.

First, a construction or repair job can be performed well or poorly. Two new variables were introduced, QP% and QS%, representing the quality of the materials and workmanship in the primary and secondary coolant loops. These variables are set equal to 1 to represent first-rate quality, to 2 to represent second-rate, and to 3 to represent third-rate. The values of QP% and QS% are selected independently of each other, to reflect the fact that sometimes one job is done well while the other is not. In running the simulation, this means that a leakage shutdown will sometimes be due to primary leakage, sometimes to secondary trouble. The values of QP% and QS% are defined on startup at statements 430 and 440, and redefined at 2020 and 2030 after a maintenance shutdown.

In selecting the values of QP% and QS%, a probability of 0.57, the square root of 0.33, is used as the threshold between first-rate and inferior quality. This means that one startup between first-rate and inferior coolant systems is first-class condition. The remaining startups will occur with one or both coolant systems impaired. As the simulation runs, QP% and QS% are used at statements 1210 and 1220 to select different damage probabilities for each coolant system.

Second, a three-step flow-dependent random damage factor has been introduced. Let's use statements 1211 and 1212 as an example. These statements define the probability of a leak occurring in the primary coolant system when the workmanship is of first-class quality. When the primary coolant flow PF% is greater than 65, there is 1 chance in 5000 of a leak occurring on any given day of operation. When the flow rate exceeds 90 the probability of a leak increases to 1 in 250, and it reaches 1 in 20 when the flow exceeds 95. With lower-quality workmanship, the flow thresholds are reduced and the damage probabilities are increased. In the primary coolant system the flow thresholds are slightly lower than in the secondary system, but the probabilites of leakage are slightly less.

This represents a sneaky pedagogical trick, introducing the idea of a rate of change in a situation where it is vitally useful.

Third, a plant aging factor has been inserted, which slowly increases the probability of a leak occurring in either cooling system as the plant accumulates operating days. These aging factors are the last terms in statements 1212, 1214, 1216, 1222, 1224, and 1226. The aging factors for poorer-quality systems have higher probabilities of leakage.

My statements 1210 to 1226 can be removed and the original statements 1210 and 1220 can be reinstated, and the PET simulation will continue to operate properly. Alternatively, users can rewrite these statements to suit their own ideas about these leakage probabilities.

GO% was used as a variable in the original program, for example in line 1380. Unfortunately, this is an illegal variable in the PET. It has been replaced with GZ%, representing turbine-generator power output.

As Berggren noted in his documentation, this program "makes extensive use of logic evaluations within expressions. For example, (A >100) equals one if true and zero if false." The PET handles such logic expressions, but returns a *negative* one if true, and zero if false. This means that all logical expressions must have their signs changed, in translating from Apple to PET Basic. To see the effect of this, compare statements 1212, 1214, etc. At first glance it would appear that these aging factors operate to reduce leakage, since they are subtracted. But the effect is additive, because we are subtracting a negative value.

In the PET version, time delays have been added to slow down the rate at which text is thrown on the screen. The time-delay subroutine starts at line 5300, and uses variable Z4 to control the time delay. Z4 is calibrated roughly in tenths of a second. In line 48 for instance, Z4 is set equal to 8. This means the next delay will be about 8 tenths of a second. Z4 is automatically incremented by 3 each time the delay routine is used. This has the effect of starting down the screen fairly quickly, then gradually slowing down as the screen fills. The delays can be adjusted by changing the Z4 coefficient in line 5300. Coefficients greater than 63 will increase the delays, less then 63 will reduce them. To prevent insanity during debugging, it is recommended that the 63 be reduced to about 6. When everything is running, restore it to a suitable value.

After this simulation run is completed, that is, when the fuel charge has been exhausted, the program offers a comment on the operator's competence. The comment is selected by calculations performed in line 4050 in the Apple version, or in lines 4020-4022 in the PET version. Additional comments have been added to the PET version, and the thresholds have been lowered, to give a bit more encouragement to the beginner. Operation at a load factor of 60% is considered to be acceptable, and a load factor of 70% is considered good. I believe these interpretations are not substantially different from those prevailing in the industry.

Finally, the screen display of operating information presented difficulties in completing the adaptation. While the Apple instructions VTAB and HTAB were easy to interpret, I couldn't figure out what CALL-958 or CALL-922 (lines 1165 and 1170) might mean. To get around this, I did a complete rewrite on this program. I can't venture an opinion whether my version is better or worse than the original, because I have no idea how the original works. However, I added a couple of features which I believe are not present in the Apple version. The PET version gives the equipment temperatures as does the Apple version, but in addition the PET

gives the change from the previous readout. This is of considerable help to the operator. The operating response to a reactor core temperature of 740 degrees will be different, depending on whether this represents a decrease of 5 degrees, an increase of 5 degrees, or an increase of 50 degrees compared to the previous temperature. This also represents a sneaky pedagogical trick, introducing the idea of a rate of change in a situation where it is vitally useful, in an attempt to foster a better understanding of the general significance of rates.

Another innovation (at least I think it is an innovation) is the continuous reporting of the leakage rates, regardless of whether they are serious enough to require shutdown. This enables the operator to judge more clearly how events are developing, and adds some dramatic suspense: "The blasted secondary leakage is up to 4 L/day, and I've just got the plant settled down at 1820 kW. How long can I continue?" Leakages have been given the physical unit of litres per day.

I have suppressed the "low power output" warning during the first eight days after a startup, to give the operator a fighting chance to get the plant under way. This suppression is performed in line 464, and uses variable D4% to identify the startup day.

In this version, the trouble warnings are presented on the screen in a frame separate from the operating conditions frame. Statement 472 suppresses the warning frame if no warnings are to be displayed, while lines 462-470 decide whether to override the suppression. The warning frame presents both warnings of out-of-range operating conditions and reports of damaged equipment. Variable SW is used to state "None" if this frame is activated but there is either no warning or no reportable damage. In the Apple version, line 4040 uses the phrase "kilowatts per day," and 770 contains a similar phrase. These have been corrected.

When an operating readout comes up on the screen, the program asks for four control settings, in turn: control rods, emergency coolant flow, primary coolant flow, and secondary coolant flow. If the current value of any of these is to remain unchanged, press the up-arrow " ↑ " without putting in a number. If all values are to remain unchanged, press ↑ four times. If a value is to be changed, type in the new value, terminated by ↑ . The input routine will automatically terminate each entry if three digits are input. As in Berggren's program, these variables are each set to a value from 0 to 100.

The Control setting input routine, statements 10-15, does not include a delete feature. This omission is deliberate, to reflect the "unforgiving" nature of

```
0 REM  NUCLEAR POWER PLANT SIMULATION
1 :
2 REM  FOR PET COMPUTER
3 :
4 REM  BY IVAN SMITH & HUME SMITH
5 :
6 REM  BASED ON S. R. BERGGREN, CREATIVE COMPUTING DECEMBER 1980 PAGE 128 FF
7 :
8 REM  REVISED 23 DECEMBER 1980
9 GOTO40
10 POKE158,0:B$="":PRINT" ";:POKE167,0
11 GETA$:IFA$=""THEN11
12 Z=ASC(A$):IFZ=94THEN15
13 IFZ<48ORZ>57THEN11
14 PRINTA$;:B$=B$+A$:IFLEN(B$)<3THEN11
15 POKE167,1:PRINT" ";:RETURN
40 POKE59468,14:R=RND(-TI):BL$="                                                                    "
41 T1$="        ▀▀▌ ▗▟▀▙▀▟▖ ▀▙▀▖ ▀▙▗▟▌"
42 T2$="▚       ▜▌▟▌▗▟ ▗▀▀▙▌▗▟ ▀▟▌"
44 PRINT"■"T1$:PRINTSPC(7)"BY  ▜▌▀▀▌▀▗ ▗▖ ▌▀▗▌▌▗▀▗":Z4=7:GOSUB5300
46 PRINTSPC(5)"■AS MODIFIED BY THE STAFF OF"
48 PRINT"■ ▌▗▖▌▀▗▟▌▀▖ ▚▌▌ ▜▀▗▛▛▖ ▗▟▀▙▟ ▜▀▌▖▟":GOSUB5300:Z4=8
50 PRINT"■■HIS PROGRAM SIMULATES THE OPERATION OF"
60 PRINT"■A NUCLEAR POWER REACTOR.";:GOSUB5300:PRINT"  HE OBJECT"
70 PRINT"■IS TO OPERATE THE PLANT AT AS HIGH A"
80 PRINT"■POWER OUTPUT AS POSSIBLE";:GOSUB5300:PRINT", WITHOUT"
90 PRINT"■CAUSING A REACTOR MELTDOWN.":GOSUB5300
92 FORZ5=1TO7:PRINTTAB(18)"■MELTDOWN■":FORZ6=1TO50:NEXTZ6
94 PRINTTAB(18)"■MELTDOWN":FORZ6=1TO50:NEXTZ6,Z5:GOSUB5310
100 PRINT"■▀O YOU WANT INSTRUCTIONS";:INPUTZ$:IFZ$="N"THEN350
102 IFZ$<>"Y"THEN100
110 PRINT"■I HE ■CONTROL RODS■ ADJUST THE AMOUNT OF"
120 PRINT"■HEAT PRODUCED BY THE REACTOR.":Z4=11:GOSUB5300
130 PRINT"■■I HE ■PRIMARY COOLANT■ TRANSFERS THIS"
140 PRINT"■HEAT TO THE HEAT EXCHANGER.":GOSUB5300
142 PRINT"■■I HE ■SECONDARY COOLANT■ PICKS UP HEAT"
144 PRINT"■FROM THE HEAT EXCHANGER";:GOSUB5300:PRINT", AND TRANSFERS"
150 PRINT"■IT TO THE ■TURBINE■";:GOSUB5300:PRINT", WHICH DRIVES"
160 PRINT"■THE ELECTRIC GENERATOR.";:GOSUB5300:PRINT"  HE SECONDARY"
170 PRINT"■COOLANT THEN GOES TO THE ■COOLING TOWER■.":GOSUB5300
172 PRINT"■AND THEN BACK TO THE HEAT EXCHANGER.":GOSUB5300:GOSUB5310
180 PRINT"■I HE ■EMERGENCY COOLANT■ IS USED TO HELP"
182 PRINT"■SHUT DOWN THE REACTOR";:Z4=11:GOSUB5300:PRINT" WHEN OTHER SYSTEMS"
184 PRINT"FAIL.":GOSUB5300:PRINT"■■▗NLIKE THE OTHER COOLANTS,"
186 PRINT"■EMERGENCY COOLANT IS NOT RECYCLED.":GOSUB5300
188 PRINT"■I HIS MEANS YOUR SUPPLY OF EMERGENCY"
190 PRINT"■COOLANT IS LIMITED.":GOSUB5300:GOSUB5310:Z4=13
192 PRINT"■▗N INCREASED FLOW OF PRIMARY COOLANT"
194 PRINT"■WILL LOWER THE REACTOR CORE TEMPERATURE":GOSUB5300
196 PRINT"■AND WILL RAISE THE HEAT EXCHANGER"
198 PRINT"■TEMPERATURE.":GOSUB5300
200 PRINT"■■▗N INCREASED FLOW OF SECONDARY COOLANT"
202 PRINT"■WILL LOWER THE HEAT EXCHANGER" PRINT"■TEMPERATURE";:GOSUB5300
204 PRINT", AND WILL INCREASE THE":PRINT"■POWER GENERATED";:GOSUB5300
206 PRINT", BUT WILL RAISE THE"
208 PRINT"■COOLING TOWER TEMPERATURE.":GOSUB5300:GOSUB5310:Z4=13
210 PRINT"■▗UCCESSFUL OPERATION DEPENDS ON"
212 PRINT"■GETTING A BALANCE";:GOSUB5300
214 PRINT", BETWEEN THE":PRINT"■HIGH TEMPERATURES NEEDED FOR"
216 PRINT"■POWER GENERATION";:GOSUB5300:PRINT" AND THE"
218 PRINT"■TEMPERATURE LIMITS OF THE MACHINERY.":GOSUB5300:GOSUB5310
220 PRINT"■■■■▀NTER  ▔  TO SEE REACTOR DIAGRAM."
222 PRINT"■▀NTER  ▖  FOR WORKING INSTRUCTIONS."
224 PRINT"■▀NTER  ▲  TO START OPERATION."
226 PRINT"■"TAB(13);:INPUTA$:IFA$="D"THENGOSUB6000:GOTO220
227 IFA$="S"THEN390
228 IFA$<>"I"THEN226
230 PRINT"■I HE CONTROLS ARE OPERATED BY TYPING IN"
235 PRINT"■THE DESIRED CONTROL ROD SETTING", Z4=13:GOSUB5300:PRINT" AND"
240 PRINT"■FLOW RATES.":GOSUB5300:PRINT"■■▗SE VALUES FROM 0 TO 100.";:GOSUB53
242 PRINT"  ▖F NO ENTRY":PRINT"■IS MADE, THE PREVIOUS SETTINGS WILL
244 PRINT"■REMAIN UNCHANGED.":GOSUB5300
250 PRINT"■■▗SE "CHR$(34)"↑"CHR$(34)" TO STEP TO THE DIFFERENT"
255 PRINT"■FUNCTIONS.";:GOSUB5300:PRINT"  OHEN THE DESIRED"
260 PRINT"■ENTRIES HAVE BEEN MADE, THE":PRINT"■NEXT DAY'S CONDITIONS";
265 PRINT" WILL BE":PRINT"■CALCULATED.":GOSUB5300:GOSUB5310
270 PRINT"■I HE REACTOR CAN BE KEPT OPERATING UNTIL"
272 PRINT"■THE REACTOR FUEL IS EXHAUSTED";:Z4=13:GOSUB5300:PRINT", OR"
274 PRINT"■UNTIL A MELTDOWN OCCURS.":GOSUB5300
276 FORZ5=1TO7:PRINTTAB(8)"■MELTDOWN■":FORZ6=1TO50 NEXTZ6
278 PRINTTAB(8)"■MELTDOWN":FORZ6=1TO50:NEXTZ6,Z5
280 PRINT"■■HE FUEL WILL LAST ABOUT 100 TO 150"
282 PRINT"■DAYS";:GOSUB5300:PRINT", DEPENDING ON HOW MUCH"
```

nuclear technology. If an operating error is made, its consequences must be endured, at least for one operating cycle. If an operator types 80 for a control rod setting when 8 was intended, the computer will give a vivid illustration of the consequences of such an error, and the operator will learn a few things as he tries to salvage the plant in subsequent cycles. I have made such errors, and have found the results to be "interesting." If a user wishes to include a delete feature, the input routine should be replaced by the following:

```
10 Same as existing 10
11 Same as existing 11
12 Z=ASC(A$):IFZ=94THEN17
13 IFZ=20ANDLEN(B$)=1THENB$
   =CHR$(20):?CHR$(20);:GOTO11:REM
   DELETE
14 IFZ=20ANDLEN(B$)=2THENB$
   =LEFT$(B$,1):?CHR$(20);:
   GOTO11
15 Same as existing 13
16 Same as existing 14
17 Same as existing 15
```

In watching students run the program, it was noted that the ↑ would sometimes be pressed five or six times, once or twice more that the required four. The extra characters remained in the keyboard input buffer, and caused the program to skip over the first one or two control inputs on the next cycle, which created unnecessary difficulties. To eliminate this, POKE 158,0 was inserted in line 10. This sets the keyboard input buffer counter to zero at the start of each input, eliminating any spurious characters which might be lurking in the buffer. The POKE 167,0 in line 10 turns the cursor on, while the POKE 167,1 in line 15 turns it off.

The POKE 216 used in lines 920, 930, 5310, etc., positions the cursor on a screen line. POKE 216,n:? positions the cursor on line n+2. For example, POKE 216,17:? places the cursor on line 19 for the next print. The POKE 216,n must be followed by a blind PRINT instruction, otherwise it won't take effect.

The PET nuclear plant simulation occupies 14.1K of memory, and presumable will work in a 16K machine. I used a 32K PET model 2001-32N.

Berggren claimed his program "is almost entirely crashproof," a statement which drew the rejoinder "famous last words." I have seen this program run more than fifty times by a variety of inexperienced operators, and have yet to see a crash. Berggren's claim seems to be justified.

I think Berggren's simulation is the most useful education program ever published by *Creative Computing*. It can form a foundation for variations which simulate other reactor types, or which go into greater detail, and it may serve as inspiration for other simulations. □

```
284 PRINT"■ENERGY IS PRODUCED EACH DAY.":GOSUB5300
290 PRINT"■■OHEN THE FUEL IS EXHAUSTED, YOUR"
295 PRINT"■PERFORMANCE WILL BE SUMMARIZED.":GOSUB5300:GOSUB5310
300 PRINT"■■F YOU WANT TO REPAIR DAMAGE OR REPLACE" PRINT"■■OOLANT", Z4=11
305 GOSUB5300:PRINT", BRING THE REACTOR TEMPERATURE"
310 PRINT"■DOWN BELOW 100";:GOSUB5300:PRINT", AND SHUT OFF THE"
315 PRINT"■COOLANT FLOWS.";:GOSUB5300:PRINT"  THIS WILL CAUSE"
320 PRINT"■AN AUTOMATIC MAINTENANCE SHUTDOWN.":GOSUB5300
325 PRINT"■■EPAIRS WILL BE MADE";:GOSUB5300:PRINT", AND ALL COOLANT"
330 PRINT"■WILL BE REPLENISHED." GOSUB5300
335 PRINT"■■HE GREATER THE DAMAGE, THE LONGER"
340 PRINT"■THE REPAIRS WILL TAKE.":GOSUB5300:GOSUB5310
350 PRINT"■"TAB(13)"■■■
```

```
744 T8%=XT%
750 PRINT"  ─OOLING TOWER  300"
752 PRINTTAB(25)"□"CT%-T9%:PRINTTAB(33)"□"CT%
754 T9%=CT%
760 PRINT"▒       ┐OWER OUTPUT (MAX. 2000):"GZ%"KO"
762 KW%=TT/DAY%:VE=INT(24*TT*.055)
764 VE$=STR$(VE):IFVE<10000THENVE$="$"+VE$:GOTO770
766 VE$="$"+LEFT$(VE$,LEN(VE$)-3)+" "+RIGHT$(VE$,3)
770 PRINT"            ♠VERAGE POWER OUTPUT:"KW%"KO"
780 PRINT"  xALUE OF ENERGY PRODUCED: "VE$
810 PRINT"▒─OOLANTS:       └─×┌  └─♠/♠┘─   ─└┌o"
820 PRINT"    ┐MERGENCY"TAB(16)EV%TAB(22)ED%"└/DAY"TAB(33)EF%
830 PRINT"      ┐RIMARY"TAB(16)PV%TAB(22)PD%"└/DAY"TAB(33)PF%
840 PRINT"    ♠ECONDARY"TAB(16)SV%TAB(22)SD%"└/DAY"TAB(33)SF%
850 IF(100-RL)<5THENPRINT"▒▒▒─EACTOR FUEL EXHAUSTED":GOTO4000
900 REM  GET NEW CONTROL VALUES
910 A2%=A1%:A1%=A%
920 POKE216,23:PRINT:PRINTBL$
930 POKE216,23:PRINT:PRINT"─ONTROL ROD POSITION?";:GOSUB10
940 IFB$=""THEN960
950 A%=VAL(B$):A%=A%+(100-A%)*(-(A%>100))
960 POKE216,23:PRINT:PRINTBL$
970 POKE216,23:PRINT:PRINT"┐MERGENCY COOLANT FLOW?";:GOSUB10
980 IFB$=""THEN1000
990 EF%=VAL(B$):EF%=EF%+(100-EF%)*(-(EF%>100))
1000 POKE216,23:PRINT:PRINTBL$
1010 POKE216,23:PRINT:PRINT"┐RIMARY COOLANT FLOW?";:GOSUB10
1020 IFB$=""THEN1040
1030 PF%=VAL(B$):PF%=PF%+(100-PF%)*(-(PF%>100))
1040 POKE216,23:PRINT:PRINTBL$
1050 POKE216,23:PRINT:PRINT"♠ECONDARY COOLANT FLOW?";:GOSUB10
1060 IFB$=""THEN1100
1070 SF%=VAL(B$):SF%=SF%+(100-SF%)*(-(SF%>100))
1100 IFPF%=0ANDSF%=0ANDRH<1ANDRT%<100ANDAX=0THEN1120
1110 GOTO1140
1120 GOSUB2000:POKE216,23:PRINT PRINTBL$
1122 POKE216,23:PRINT:PRINT"\AINTENANCE SHUTDOWN:"MD%"DAYS.□"
1130 FORI=1TO5000:NEXTI
1140 IFEF%>EV%THENEF%=EV%
1200 REM  DAMAGE ASSESSMENT AND OPERATION CALCULATIONS
1205 EV%=EV%-EF%+2*ED%*(ED%>3)
1210 ON@P%GOTO1211,1213,1215 REM LEAKAGE PROBABILITIES
1211 PD%=PD%+(PF%>65)*(RND(2)>.9998)+(PF%>90)*(RND(2)>.996)
1212 PD%=PD%+(PF%>95)*(RND(2)>.95)-(4000*RND(2)<DAY%):GOTO1220
1213 PD%=PD%+(PF%>55)*(RND(2)>.9995)+(PF%>80)*(RND(2)>.993)
1214 PD%=PD%+(PF%>92)*(RND(2)>.94)-(2700*RND(2)<DAY%):GOTO1220
1215 PD%=PD%+(PF%>40)*(RND(2)>.999)+(PF%>70)*(RND(2)>.985)
1216 PD%=PD%+(PF%>87)*(RND(2)>.93)-(1400*RND(2)<DAY%)
1220 ON@S%GOTO1221,1223,1225
1221 SD%=SD%+(SF%>75)*(RND(2)>.9997)+(SF%>93)*(RND(2)>.995)
1222 SD%=SD%+(SF%>97)*(RND(2)>.94)-(3000*RND(2)<DAY%):GOTO1230
1223 SD%=SD%+(SF%>60)*(RND(2)>.9992)+(SF%>85)*(RND(2)>.99)
1224 SD%=SD%+(SF%>94)*(RND(2)>.92)-(2400*RND(2)<DAY%):GOTO1230
1225 SD%=SD%+(SF%>40)*(RND(2)>.9985)+(SF%>78)*(RND(2)>.98)
1226 SD%=SD%+(SF%>89)*(RND(2)>.9)-(1800*RND(2)<DAY%)
1230 PB%=-(PD%>5)
1240 SB%=-(SD%>5)
1250 IFPF%>(100-PD%*10)ANDPB%>0THENPF%=(100-PD%*10)*(-(100-PD%*10>0))
1260 IFSF%>(100-SD%*10)ANDSB%>0THENSF%=(100-SD%*10)*(-(100-SD%*10>0))
1270 RL=RL+RH/50
1280 RH=(A%*30+A1%*60+A2%*10)/2500*(100-RL)
1300 PH=PF%*(100*(-(PV%>100))+PV%*(-(PV%<=100)))/350
1310 EH=EF%/200*(RT%-25)
1320 RT%=RT%+RH-EH-PH-5*(-(RT%>25))
1325 RT%=25+(RT%-25)*(-(RT%>25))
1330 XT%=((RT%-25)*PF%+(CT%-25)*SF%)/(PF%+SF%+1)+25
1340 IFXB%>0THENXT%=RT%*.8+5
1350 SH=SF%*(100*(-(PV%>100))+PV%*(-(PV%<=100)))/350*(XT%-CT%)
1360 IFXB%>0THENSH=SH*.2
1370 GZ%=SH/XT%*(XT%-CT%)*2/3
1375 IFGZ%>2600THENGZ%=2600
1380 GZ%=GZ%*(-(GZ%>0))*(-(GB%=0))
1390 CT%=25+((XT%-25)*(SH-GZ%)/(SH+1)*.75)
1395 CT%=25*(-(CT%<=25))+CT%*(-(CT%>25))
1400 IFXB%<1THENXB%=(-(XD%>2))*(-(RND(4)>.9))
1410 IFGB%<1THENGB%=(-(GD%>4))*(-(RND(4)>.9))
1420 TT=TT+GZ%
1430 GOTO460
2000 REM  MAINTENANCE REPAIR SUBROUTINE
2010 EV%=300:PV%=120:SV%=120
2020 QP%=1-(RND(2)>.57)-(RND(2)>.95)
2030 QS%=1-(RND(2)>.57)-(RND(2)>.95)
2040 RT%=25:T7%=25
2050 XT%=25:T8%=25
2060 CT%=25:T9%=25
2070 DMGE%=DMGE%+2*RD%+ED%+PD%+XD%+SD%+GD%
2080 MD%=5+3*(10*(-(RD%>3))-(ED%>3)-(PD%>3)-(SD%>3)+2*PB%+2*SB%+3*XB%+3*GB%)
```

```
2082 DAY%=DAY%+MD%:D4%=DAY%
2090 RD%=0
2100 ED%=0
2110 PD%=0
2120 XD%=0
2130 SD%=0
2140 GD%=0
2150 PB%=0
2160 SB%=0
2170 XB%=0
2180 GB%=0
2190 EF%=0:PF%=0:SF%=0
2195 GZ%=0
2200 RETURN
3000 PRINT"■■■■■":REM  MELTDOWN ENDING
3010 IFRD%>6GOTO3100
3015 PRINT"■  THE REACTOR CORE HAS BEEN DESTROYED"
3020 PRINT"BY UNCONTROLLED THERMAL RUNAWAY."
3025 PRINT" HOWEVER, THE CONTAINMENT BUILDING HAS"
3030 PRINT"NOT YET RUPTURED."
3035 PRINT"■  INITIATE YOUR EVACUATION PLAN.":GOTO3200
3100 PRINT"■  THE REACTOR CORE HAS MELTED DOWN AND"
3105 PRINT"PRODUCED A STEAM EXPLOSION. THE"
3110 PRINT"CONTAINMENT BUILDING HAS RUPTURED."
3115 PRINT"LETHAL RADIOACTIVE GASSES AND DEBRIS"
3120 PRINT"HAVE ESCAPED."
3125 PRINT"■  INITIATE YOUR EVACUATION AND"
3130 PRINT"RADIATION CLEANUP PLANS AND GET MEDICAL"
3135 PRINT"ASSISTANCE."
3200 PRINT"■■■"
3210 FORZ5=1TO30
3220 PRINT"■    ▚▞▙▟▜▛▚▞▙▟▜▛    ▚▞▙▟▜▛▚▞▙▟▜▛■    ▚▞▙▟▜▛▚▞▙▟▜▛"
3230 FORZ6=1TO100-3*Z5:NEXTZ6
3240 PRINT"■    ▚▞▙▟▜▛▚▞▙▟▜▛■    ▚▞▙▟▜▛▚▞▙▟▜▛    ▚▞▙▟▜▛▚▞▙▟▜▛■"
3250 FORZ6=1TO100-3*Z5:NEXTZ6
3260 NEXTZ5
3270 POKE216,21:PRINT:GOTO5000
4000 REM EVALUATION OF GAME
4005 PRINT"■  OVER A PERIOD OF"DA%"DAYS, YOU HAVE"
4010 PRINT"PRODUCED AN AVERAGE POWER OUTPUT OF"
4015 PRINT"■■"KW%"KW."
4020 AK%=1-(KW%>799)-(KW%>899)-(KW%>999)-(KW%>1099)-(KW%>1199)-(KW%>1299)
4022 AK%=AK%-(KW%>1399)-(KW%>1499)
4025 PRINT"■   OUR AVERAGE POWER PRODUCTION RATE IS"
4030 ONAK%GOTO4032,4034,4036,4038,4040,4042,4044,4050,4055
4032 PRINT"HORRIBLE! FIND A LESS DEMANDING JOB.":GOTO4060
4034 PRINT"BAD.  OUR AREA HAD":PRINT"FREQUENT BLACKOUTS.":GOTO4060
4036 PRINT"PRETTY BAD.  OUR AREA HAD":PRINT"OCCASIONAL BLACKOUTS.":GOTO4060
4038 PRINT"WAY BELOW YOUR AREA'S POWER NEEDS.":GOTO4060
4040 PRINT"NOT GOOD.   OUR AREA HAD BROWNOUTS.":GOTO4060
4042 PRINT"NOT BAD.   OUR AREA'S POWER NEEDS"
4043 PRINT"WERE SUPPLIED - JUST.":GOTO4060
4044 PRINT"ADEQUATE.  OU COULD DO BETTER.":GOTO4060
4050 PRINT"EXELLENT! POWER RATES IN YOUR AREA WILL"
4051 PRINT"NOT BE INCREASED.":GOTO4060
4055 PRINT"NEAR THE MAXIMUM! POWER COSTS IN YOUR"
4056 PRINT"AREA WILL DROP SIGNIFICANTLY."
4060 GOSUB2000
4065 D%=1-(DM%>10)-(DM%>20)-(DM%>30)
4070 PRINT"■  THE EQUIPMENT DAMAGE SUSTAINED WAS"
4075 ONDM%GOTO4080,4085,4090,4095
4080 PRINT"VERY LIGHT.":GOTO5000
4085 PRINT"MODERATE.":GOTO5000
4090 PRINT"SERIOUS." GOTO5000
4095 PRINT"SEVERE."
5000 REM END
5005 INPUT"■WOULD YOU LIKE TO TRY AGAIN";A$:A$=LEFT$(A$,1)
5010 IFA$="Y"THENGOSUB2000:GOTO390
5015 PRINT"■":END
5100 B$=""
5105 FORI=1TO4
5110 GETA$
5130 IFASC(A$)<>94THEN5140
5134 IFB$=""THENRETURN
5136 OUT%=VAL(B$)
5138 RETURN
5140 IFASC(A$)=13THENSYS50583:GOTO1170
5150 B$=B$+A$
5160 PRINTA$;
5170 NEXTI
5180 RETURN
5300 FORJ6=1TO63*Z4:NEXTJ6:Z4=Z4+3:RETURN:REM DELAY
5310 POKE216,23:PRINT:PRINT"     PRESS ● ▚▞▙▟▜▛ TO CONTINUE...■";
5311 WAIT59410,4,4:POKE216,23:PRINT:PRINTBL$:GETZZ$:RETURN
6000 ZD$="■■■■■
```

```
6010 PRINT"
6015 PRINT"
6020 FORZZ=1TO4:PRINT"
6025 PRINT"
6030 PRINT"
6035 PRINT"
6040 Z4=11:GOSUB5300:GOSUB5310
6050 PRINTZD$"
6055 PRINT"
6060 FORZZ=1TO3:PRINT"
6065 GOSUB5300:GOSUB5310
6070 PRINTZD$"
6075 PRINT"
6080 PRINT"
6085 PRINT",
6086 PRINT"
6090 PRINT"
6095 PRINT"
6096 PRINT"
6097 FORZZ=1TO3:PRINT"
6100 GOSUB5300:GOSUB5310
6105 PRINTZD$"
6110 PRINT"
6115 PRINTTAB(5)"
6120 PRINTTAB(5)"
6125 PRINTTAB(5)"
6126 PRINTTAB(5)"
6130 Z4=11:GOSUB5300:GOSUB5310
6135 PRINTZD$"
6140 PRINT"
6150 PRINTTAB(12)"
6155 PRINTTAB(13)"
6156 PRINTTAB(13)"
6160 PRINTTAB(8)"
6165 PRINTTAB(15)"
6170 PRINTTAB(8)"
6175 GOSUB5300:GOSUB5310
6180 PRINTZD$"
6185 PRINT"
6190 PRINTTAB(14)"
6195 PRINTTAB(14)"
6200 PRINTTAB(14)"
6205 PRINTTAB(14)"
6210 Z4=11 GOSUB5300:GOSUB5310
6220 PRINTZD$"
6225 PRINT"
6230 PRINTTAB(17)"
6235 PRINTTAB(18)"
6236 PRINTTAB(18)"
6240 PRINTTAB(16)"
6250 PRINTTAB(16)"
6255 PRINTTAB(16)"
6260 PRINTTAB(18)"
6265 PRINTTAB(18)"
6270 PRINTTAB(33)"
6271 PRINTTAB(33)"
6275 PRINTTAB(33)"
6280 PRINTTAB(21)"
6281 PRINTTAB(21)"
6285 GOSUB5300:GOSUB5310
6290 PRINTZD$"
6295 PRINT"
6300 PRINTTAB(17)"
6305 PRINTTAB(17)"
6310 PRINTTAB(17)"
6315 GOSUB5300:GOSUB5310
6320 PRINTZD$"
6325 PRINT"
6330 PRINTTAB(10)"
6335 PRINTTAB(10)"
6340 PRINTTAB(10)"
6345 Z4=11:GOSUB5300:GOSUB5310
6350 PRINTZD$"
6355 PRINT"
6360 PRINTTAB(24)"
6365 PRINTTAB(25)"
6370 PRINTTAB(26)"
6371 PRINTTAB(26)"
6372 PRINTTAB(26)"
6375 PRINTTAB(25)"
6380 PRINTTAB(25)"
6385 GOSUB5300:GOSUB5310
6390 PRINTZD$"
6395 GOSUB5310:PRINT"
```

Geography

Jonathan Amsterdam

Geography is a simple two-player game which has been adapted to a game against the computer. The instructions in the sample run explain how to play the game.

The only problem in writing a computer program for Geography was the list of geographical names. In the actual person vs. person game, any geographical location can be used. I chose to restrict possible choices to the categories listed in the instructions. This results in about three hundred pieces of data—sufficient for a challenging game but not enough to appreciably slow down the computer. The list can, of course, be extended by including categories such as world's highest mountains, world's longest rivers, state counties, etc. If the game begins to slow down, you could arrange the data in alphabetical order and use a more efficient search, for example, a binary search. I use a sequential search, with the advantage of being able to add new data right onto the end of the program.

Geography was written on a PDP-11/70 with RSTS/E BASIC. String functions may present the only problem in translation. In RSTS/E BASIC, for some reason, LEFT(A$,N) will take the N leftmost characters of A$, but RIGHT(A$,N) will take the right part of A$ starting from the Nth character.

Important variables: A is the number of data elements; when adding data, it is necessary only to change the value of A (line 30) and the DIM statement (line 40). Everything else will take care of itself. B$ holds the data; U$ holds the items already used in the game; U is the counter for U$ and C indicates whose turn it is.

Most of the code is self-explanatory; the only confusing part might be the search in lines 400-500. To prevent the program from choosing the same response in every game, the search for a match begins at a random point in the middle of the array, proceeds to the end, then goes from the beginning to the original random point. If no match is found, the player wins; otherwise, the response is printed and it is the player's turn.

Finally, for those lucky few with voice synthesizers, Geography would be an ideal application of computer speech. □

Jonathan Amsterdam, 848 Hayes St., Baldwin, NY 11510.

```
INSTRUCTIONS? YES
THIS IS A GAME OF 'GEOGRAPHY.'  ONE PLAYER NAMES A
GEOGRAPHICAL LOCATION.  THE OTHER PLAYER MUST NAME
A LOCATION WHOSE FIRST LETTER IS THE SAME AS THE
LAST LETTER OF THE PRECEEDING WORD.  YOU WILL PLAY
AGAINST THE COMPUTER.  ONLY THE FOLLOWING
CATEGORIES WILL BE USED:

                        CONTINENTS
                        COUNTRIES
                        U.S. STATES
                        CANADIAN PROVINCES
                        MAJOR WORLD CITIES
                        U.S. STATE CAPITALS
                        OCEANS AND SEAS

THE COMPUTER WILL TELL YOU IF THE LETTERS OF
YOUR ENTRY AND THE PRECEEDING WORD DON'T MATCH,
THE WORD HAS ALREADY BEEN USED, OR THE WORD IS
NOT IN THE COMPUTER'S MEMORY ('ILLEGAL WORD').
IF YOU CAN'T THINK OF A CORRECT RESPONSE,
YOU MAY TYPE 'CHALLENGE.'  IF YOU'VE REALLY
EXHAUSTED ALL POSSIBLE RESPONSES, YOU'VE WON. OTHERWISE,
THE COMPUTER WILL PRINT OUT A POSSIBLE RESPONSE, AND
YOU LOSE.
DO YOU WANT TO GO FIRST? NO
                                ATLANTA
ZAIRE                           ? AUSTRIA
? EAST GERMANY                  ALBANY
YEMEN                           ? YUGOSLAVIA
? NEW YORK                      AUSTIN
KENYA                           ? NORTH AMERICA
? ALABAMA                       ATLANTIC OCEAN
ATHENS                          ? NORTH DAKOTA
? SALT LAKE CITY                ARCTIC OCEAN
YELLOW SEA                      ? NORTH CAROLINA
? ASIA                          ARKANSAS
AMSTERDAM                       ? SOUTH DAKOTA
? MISSISIPPI                    ARIZONA
*** ILLEGAL WORD ***            ? ALBANY
? MISSISSIPPI                   *** ALREADY USED ***
INDIANA                         ? CHALLENGE
? AUGUSTA                       ALASKA
ANNAPOLIS                       YOU LOSE
? SOUTH CAROLINA                PLAY AGAIN? NO
```

```
 1 REM  **********************
 2 REM  *      GEOGRAPHY      *
 3 REM  *                     *
 4 REM  * JONATHAN AMSTERDAM  *
 5 REM  *  848 HAYES STREET   *
 6 REM  *  BALDWIN NY 11510   *
 7 REM  *                     *
 8 REM  *     JUNE 1979       *
 9 REM  **********************
20 RANDOMIZE
30 A=345
40 DIM B$(345),U$(345)
50 FOR I=1 TO A\READ B$(I)\NEXT I
60 PRINT "INSTRUCTIONS";\INPUT A$\IF A$<>"YES" THEN 230
70 PRINT "THIS IS A GAME OF 'GEOGRAPHY.'  ONE PLAYER NAMES A"
80 PRINT "GEOGRAPHICAL LOCATION.  THE OTHER PLAYER MUST NAME"
90 PRINT "A LOCATION WHOSE FIRST LETTER IS THE SAME AS THE"
100 PRINT "LAST LETTER OF THE PRECEEDING WORD.  YOU WILL PLAY"
110 PRINT "AGAINST THE COMPUTER.  ONLY THE FOLLOWING "
```

```
120 PRINT "CATEGORIES WILL BE USED:"
130 PRINT\FOR I=1 TO 7\READ C$\PRINT TAB(25);C$\NEXT I\PRINT
140 PRINT "THE COMPUTER WILL TELL YOU IF THE LETTERS OF"
150 PRINT "YOUR ENTRY AND THE PRECEEDING WORD DON'T MATCH,"
160 PRINT "THE WORD HAS ALREADY BEEN USED, OR THE WORD IS"
170 PRINT "NOT IN THE COMPUTER'S MEMORY ('ILLEGAL WORD')."
180 PRINT "IF YOU CAN'T THINK OF A CORRECT RESPONSE,"
190 PRINT "YOU MAY TYPE 'CHALLENGE.'  IF YOU'VE REALLY"
200 PRINT "EXHAUSTED ALL POSSIBLE RESPONSES, YOU'VE WON. OTHERWISE,"
210 PRINT "THE COMPUTER WILL PRINT OUT A POSSIBLE RESPONSE, AND"
220 PRINT "YOU LOSE."
230 U=0\C=0\PRINT "DO YOU WANT TO GO FIRST";\INPUT A$
240 PRINT\IF A$="NO" THEN 250
245 C=1\GOTO 260
250 R$ CHR$(INT(RND*26+65))\GOTO 400
255 C=0
260 INPUT A$\IF C=1 THEN 300
270 IF A$="CHALLENGE" THEN 550
280 IF LEFT(A$,1)=R$ THEN 300
290 PRINT "*** LETTERS DON'T MATCH ***"\GOTO 260
300 FOR E=1 TO U
310     IF A$=U$(E) THEN 330
320     PRINT "*** ALREADY USED ***"\GOTO 260
330 NEXT E
340 FOR E=1 TO A
350     IF A$=B$(E) THEN 370
360     GOTO 390
370 NEXT E
380 PRINT "*** ILLEGAL WORD ***"\GOTO 260
390 R$=RIGHT(A$,LEN(A$))\U=U+1\U$(U)=A$
400 G=INT(RND*A+1)\H=A
410 FOR E=G TO H
420     IF LEFT(B$(E),1)=R$ THEN 480
430     FOR I=1 TO U
440             IF B$(E)=U$(I) THEN 480
450     NEXT I
460     PRINT B$(E)\U=U+1\U$(U)=B$(E)
470     R$=RIGHT(B$(E),LEN(B$(E)))\GOTO 255
480 NEXT E
490 IF G=1 THEN 510
500 H=G\G=1\GOTO 410
510 PRINT "YOU WIN"
520 PRINT "PLAY AGAIN";
530 INPUT A$\IF A$="YES" THEN 230
540 GOTO 1350
550 REM *** CHALLENGE SUBROUTINE ***
560 FOR E=1 TO A
570     IF LEFT(B$(E),1)=R$ THEN 620
580     FOR I=1 TO U
590             IF B$(E)=U$(I) THEN 620
600     NEXT I
610     PRINT B$(E)\PRINT "YOU LOSE"\GOTO 520
620 NEXT E\GOTO 510
630 DATA "NEW YORK","NEW JERSEY","PENNSYLVANIA","CONNECTICUT","ALABAMA"
640 DATA "VERMONT","NEW HAMPSHIRE","MAINE","FLORIDA","MASSACHUSETTS"
650 DATA "GEORGIA","SOUTH CAROLINA","RHODE ISLAND","UTAH","INDIANA"
660 DATA "NORTH CAROLINA","VIRGINIA","SOUTH DAKOTA","OHIO","OREGON"
670 DATA "NEW MEXICO","NORTH DAKOTA","WISCONSIN","MICHIGAN","IDAHO"
680 DATA "CALIFORNIA","MISSISSIPPI","KANSAS","NEBRASKA","IOWA"
690 DATA "WASHINGTON","MONTANA","MINNISOTA","KENTUCKY","TENNESSEE"
700 DATA "LOUISIANA","DELAWARE","MARYLAND","ARKANSAS","COLORADO"
710 DATA "ARIZONA","WEST VIRGINIA","TEXAS","NEVADA","OKLAHOMA"
720 DATA "WYOMING","MISSOURI","ALASKA","HAWAII","ILLINOIS"
730 DATA "ALBERTA","MANITOBA","SASKATCHEWAN","QUEBEC","ONTARIO"
740 DATA "PRINCE EDWARD ISLAND","NEW BRUNSWICK","NEWFOUNDLAND"
750 DATA "AFGHANISTAN","AFRICA","ALBANIA","ALGERIA","ANDORRA"
760 DATA "ANGOLA","ANTARCTICA","ARGENTINA","ASIA","AUSTRALIA"
770 DATA "AUSTRIA","BAHRAIN","BANGLADESH","BARBADOS","BELGIUM"
780 DATA "BHUTAN","BOLIVIA","BOTSWANA","BRAZIL","BRITISH HONDURAS"
790 DATA "BULGARIA","BURMA","BURUNDI","CAMBODIA","CAMEROON"
800 DATA "CANADA","CENTRAL AFRICAN REPUBLIC","CEYLON","CHAD","CHILE"
810 DATA "CHINA","COLUMBIA","CONGO","COSTA RICA","CUBA"
820 DATA "CYPRUS","CZECHOSLOVAKIA","DAHOMEY","DENMARK","DOMINICAN REPUBL
IC"
830 DATA "ECUADOR","EGYPT","EL SALVADOR","ENGLAND","ETHIOPIA"
840 DATA "EQUATORIAL GUINEA","EUROPE","FIJI","FINLAND","FRANCE"
850 DATA "GABON","GAMBIA","EAST GERMANY","WEST GERMANY","GERMANY"
860 DATA "GHANA","GREAT BRITAIN","GREECE","GUATEMALA","GUINEA"
870 DATA GUYANA,HAITI,HOLLAND,HONDURAS,HUNGARY
880 DATA "ICELAND","INDIA","INDONESIA","IVORY COAST","IRAN"
890 DATA "IRAQ","IRELAND","ISRAEL","ITALY","JAMAICA"
900 DATA "JAPAN","JORDAN","KENYA","SOUTH KOREA","NORTH KOREA"
910 DATA "KUWAIT","LAOS","LEBANON","LESOTHO","LIBERIA"
920 DATA "LIBYA","LIECHTENSTEIN","LUXEMBOURG","MALAWI","MALAYA"
930 DATA "MALAGASY REPUBLIC","MALAYSIA","MALDIVES","MALI","MALTA"
940 DATA "MAURITANIA","MAURITIUS","MEXICO","MONGOLIA","MOZAMBIQUE"
950 DATA "NEPAL","NETHERLANDS","NEW ZEALAND","NICARAGUA","NIGER"
960 DATA "NIGERIA","NORWAY","OMAN","PAKISTAN","PANAMA"
970 DATA "PARAGUAY","PERU","PHILIPPINES","POLAND","PORTUGAL"
980 DATA "QATAR","RHODESIA","RUMANIA","RWANDA","SAUDI ARABIA"
990 DATA "SCOTLAND","SENEGAL","SIERRA LEONE","SINGAPORE","SOMALIA"
1000 DATA "SOUTH AFRICA","SOUTH AMERICA","NORTH AMERICA","SPAIN","SUDAN"
1010 DATA "SURINAM","SWAZILAND","SWEDEN","SWITZERLAND","SYRIA"
1020 DATA "TANZANIA","THAILAND","TOGO","TRINIDAD","TUNISIA"
1030 DATA "UNION OF SOVIET SOCIALIST REPUBLICS","WALES","ZAIRE","YEMEN",
"ZAMBIA"
1040 DATA "UPPER VOLTA","URUGUAY","VATICAN CITY","VENEZUELA","UNITED KIN
GDOM"
1050 DATA "NORTH VIETNAM","SOUTH VIETNAM","RUSSIA","UNITED STATES","YUGO
SLAVIA"
1060 DATA "TOKYO","LONDON","SHANGHAI","OSAKA","MOSCOW"
1070 DATA "BUENOS AIRES","PARIS","MEXICO CITY","CHICAGO","LOS ANGELES"
1080 DATA "CALCUTTA","PEKING","SAO PAULO","BOMBAY","DJAKARTA"
1090 DATA "SEOUL","RIO DE JANEIRO","CAIRO","PHILADELPHIA","DETROIT"
1100 DATA "HONG KONG","BERLIN","LENINGRAD","MANILA","NEW DEHLI"
1110 DATA "BOSTON","KARACHI","MADRID","SAN FRANCISCO","WASHINGTON DC"
1120 DATA "SYDNEY","ROME","MILAN","SANTIAGO","MONTREAL"
1130 DATA "LIMA","BARCELONA","MELBOUNE","PITTSBURG","MANCHESTER"
1140 DATA "ST. LOUIS","TORONTO","HAMBURG","BUDAPEST","CLEVELAND"
1150 DATA "BANGKOK","JOHANNESBURG","ATHENS","SINGAPORE","ISTANBUL"
1160 DATA "VIENNA","BRUSSELS","AMSTERDAM","GLASGOW","HAVANA"
1170 DATA "LISBON","WARSAW","COPENHAGEN","STOCKHOLM","PRAGUE"
1180 DATA "MONTGOMERY","JUNEAU","PHOENIX","LITTLE ROCK","SACRAMENTO"
1190 DATA "DENVER","HARTFORD","DOVER","TALLAHASSEE","ATLANTA"
1200 DATA "HONOLULU","BOISE","SPRINGFIELD","INDIANAPOLIS","DES MOINES"
1210 DATA "TOPEKA","FRANKFORT","BATON ROUGE","AUGUSTA","ANNAPOLIS"
1220 DATA "LANSING","ST. PAUL","JACKSON","JEFFERSON CITY","HELENA"
1230 DATA "LINCOLN","CARSON CITY","CONCORD","TRENTON","SANTA FE"
1240 DATA "ALBANY","RALEIGH","BISMARCK","COLUMBUS","OKLAHOMA CITY"
1250 DATA "SALEM","HARRISBURG","PROVIDENCE","COLUBMIA","PIERRE"
1260 DATA "NASHVILLE","AUSTIN","SALT LAKE CITY","MONTPELIER","RICHMOND"
1270 DATA "OLYMPIA","CHARLESTON","MADISON","CHEYENNE","PACIFIC OCEAN"
1280 DATA "ATLANTIC OCEAN","INDIAN OCEAN","ARCTIC OCEAN","RED SEA","BLAC
K SEA"
1290 DATA "MEDITERRANEAN SEA","BERING SEA","NORTH SEA","BALTIC SEA","YEL
LOW SEA"
1300 DATA "CARIBBEAN SEA","GULF OF MEXICO","EAST CHINA SEA","HUDSON BAY"
1310 DATA "SEA OF JAPAN","SOUTH CHINA SEA","XENIA","XANTHUS"
1320 DATA "YALTA","YANGTZE RIVER","YOKOHAMA","YUKON TERRITORY"
1330 DATA "CONTINENTS","COUNTRIES","U.S. STATES","CANADIAN PROVINCES"
1340 DATA "MAJOR WORLD CITIES","U.S. STATE CAPITALS","OCEANS AND SEAS"
1350 END
```

CMAPS: A Basic Language Program for Chloropleth Mapping

James W. Cerny

The best synonym for the cartographer's term "choropleth map" would be "shading-by-area map". This is the familiar type of map in which there is a single data value for each observation area (state, county, census tract, etc.). Typically this involves arraying our data in ascending order so we can classify it into data classes or intervals, with a different shading symbol assigned to each data class. The map is constructed by taking each data value, looking up the appropriate data class, and shading the area with the associated shading symbol for that data class. It is a conservative mapping technique, because we assume the data values apply uniformly throughout each observation area, making no inferences about gradients. A general introduction to this and other types of mapping can be found in Robinson and Sale (1969).

M. W. Scripter (1969) devised a Fortran computer program for choropleth mapping on computers with limited storage. Since then, many Fortran versions of his program have been produced (Caspall and Jozwiak, 1973; Whitmore, 1972). Listed and discussed below, however, is CMAPS, a Basic-language version of Scripter's program designed for interactive computing on a terminal. The program, as presented, is only of moderate length and complexity and many embellishments are possible.

Fundamental to computer mapping is some method of coding information to describe the base area to be mapped. CMAPS uses a scan-line technique as follows. First, you must select a base map of a suitable scale. The areas on this base map will be approximated using the print symbols available on the terminal. Since most terminals print (or display) 10 characters per inch horizontally and 6 characters per inch vertically, you should overlay the base map with a 6 x 10 to the inch grid. Each row in this grid will be one row in the printed map. Obviously very small areas will be difficult to represent without a map of sufficiently large scale. Each row will be composed of segments, one segment representing a slice of a map area or the background surrounding the map. Map areas should be numbered in an arbitrary but consistent manner, either from top to bottom or from an alphabetical list of the areas. The background is assigned the number N+1, where N is the number of map areas. Each segment requires two pieces of information to describe it to the CMAPS program: the number of the map area it represents and the rightmost grid cell (print position) it occupies in that scan-line row. A whole scan-line data file is thus constructed for each base area that will be mapped. Each scan-line in that file consists of the number of segments in that line and then, in pairs, the segment number and rightmost print position. The leftmost print positions are deduced by CMAPS. You should also realize that to change the size of the printed map requires construction of a new scan-file.

The version of CMAPS listed here (Figure 1) allows an unlimited number of scan-lines or length of the map, up to 20 map areas (not counting background), up to 20 data classes, and will produce a map up to 72 characters wide. Before running CMAPS a scan-line file as described above must have been created. Then the program asks for a map title, map parameters, data class limits, print symbols for the data classes, and the data itself. In the sample run shown (Figure 2), a scan-line file of New Hampshire was used that was 50 scan-lines long, 45 columns wide, and composed of 10 map areas or counties (Figure 3). The data mapped were the percentage of the 1970 population in each county that was either first or second generation Canadian. The county order for numbering in the scan deck, from 1 through 10, was Coos, Grafton, Carroll, Belknap, Sullivan, Merrimack, Strafford, Rockingham, Cheshire, and Hillsborough. The program operates as follows. It classifies the data according to the data values and class limits read in. If a value is on the border between two classes, it is assigned to the lower data class. Then CMAPS begins a loop, reading information from the scan file one line at a time. It never stores more than one line of the map at a time. It reads how many segments there are for that line, and for each segment reads which map area that is, deduces which contiguous print positions belong to that segment, computes which map symbol should be associated with the segment on the basis of the data classification already done, proceeding until it has composed a whole line, which is then printed.

In the sample run (Figure 2) the highest values are in the northernmost county (Coos) and in the counties where there was the greatest industrialization in the nineteenth century. Most of the people of Canadian origin in New Hampshire are of French background, drawn to the prospect of work in the mills and factories when a different economic situation existed. Note that an attempt has been made to use darker shading for higher values or percentages. A different map classification might show more or less detail to the pattern.

In our implementation of CMAPS, we use a separate program to print an explanatory description of CMAPS and choropleth mapping, but such information could be incorporated directly in the CMAPS program. We also have a little utility program called SCANMAKE (Figure 4) to facilitate the construction of scan files by people who are novices. CMAPS could be made more retentive, storing shading and data values so that users would only have to supply new data class limits to experiment in the mapping of a set of data. At the risk of greatly enlarging memory requirements, the scan-file information could be read in and stored in an array, or it could be stored in data statements. In making modifications perhaps two things should be kept in mind. People are not likely to seek finished quality maps from CMAPS, but it can be very useful for working maps and quick proofing of data classifications. It is time consuming to print large numbers of maps on the same base at a terminal and perhaps that type of production work is better done in a batch environment.

References

Caspall, F. C., and Jozwiak, L. 1973. "CORMAP: A Choroplethic Mapping Routine," *Technical Report No. 1,* Institute for Regional, Rural, and Community Studies, Western Illinois University.

Robinson, A., and Sale, R. 1969. *Elements of Cartography,* 3rd ed. New York: Wiley & Sons.

Scripter, M. W. 1969. "Choropleth Maps on Small Digital Computers," *Proceedings,* Association of American Geographers, v. 1, 133–136.

Whitmore, S. E. 1972. "INMAP: A Computer-Aided Instruction Program for Teaching Class-Interval Selection," pp. 39–42 in *Proceedings,* Conference on Computer Applications in Geography, 12–14 October 1972, Department of Geography, SUNY at Albany.

```
100 REM CMAPS . . . CHOROPLETH MAPPING PROGRAM.
110 REM JAMES W. CERNY, MARCH 1975.
120 REM REFERENCES: (1) SCRIPTER, M.W.1969. "CHOROPLETH MAPS ON
130 REM SMALL DIGITAL COMPUTERS," PROCEEDINGS, ASSOCIATION OF
140 REM AMERICAN GEOGRAPHERS, V. 1, 133-136.  (2) CASPALL, F.C. AND
150 REM JOZWIAK, L. 1973. "CORMAP: A CHOROPLETHIC MAPPING ROUTINE,"
160 REM TECHNICAL REPORT NO. 1, INSTITUTE FOR REGIONAL, RURAL, AND
170 REM COMMUNITY STUDIES, WESTERN ILLINOIS UNIVERSITY.
180 DIM A(72),B(72),C$(22),D(20),E$(72),U(21),X(20)
190 PRINT "PROGRAM CMAPS MAKES CHOROPLETH MAPS.  SEE **CMAPSDOC."
200 PRINT
210 PRINT "ENTER THE MAP TITLE (UP TO 18 CHARS)."
220 INPUT F$
230 PRINT "ENTER THE NO. OF CLASSES, AREAS, SCAN LINES,"
240 PRINT "AND MAP WIDTH IN COLUMNS."
250 INPUT N1,N2,N3,L
260 N4=N1+1
270 PRINT "ENTER THE LOWER LIMIT FOR THE FIRST CLASS AND,"
280 PRINT "IN ORDER, THE UPPER LIMIT FOR EACH CLASS."
290 MAT INPUT U(N4)
300 U(N4)=U(N4)*.00001+U(N4)
310 REM ASSIGN BACKGROUND AND OUT-OF-RANGE SYMBOLS.
320 C$(N1+1)='.'
330 C$(N1+2)='R'
340 PRINT "ENTER A PRINT SYMBOL FOR EACH CLASS, ONE AT A TIME,"
350 PRINT "IN ORDER.  THE SYMBOLS '.' AND 'R' ARE RESERVED."
360 FOR M=1 TO N1
370 INPUT C$(M)
380 NEXT M
390 PRINT "ENTER THE DATA TO BE MAPPED, ONE VALUE FOR EACH"
400 PRINT "MAP AREA.  THE ORDER MUST MATCH SCAN DECK NUMBERING."
410 MAT INPUT X(N2)
420 PRINT "WHAT IS THE NAME OF YOUR SCAN-LINE FILE?"
430 INPUT G$
440 OPEN 1,G$,INPUT
450 REM CLASSIFY MAP AREAS ACCORDING TO DATA VALUES.
460 D(N2+1)=N4
470 FOR M=1 TO N2
480 IF X(M)< U(1) THEN 530
490 FOR N=2 TO N4
500 Q=N-1
510 IF U(N) > X(M) THEN 540
520 NEXT N
530 Q=N1+2
540 D(M)=Q
550 NEXT M
560 REM BEGIN SEQUENCE TO COMPOSE AND PRINT MAP LINES.
570 FOR N=1 TO 5
580 PRINT
590 NEXT N
600 PRINT F$
610 PRINT
620 FOR M=1 TO N1
630 PRINT "CLASS";M;"SYMBOL ";C$(M);"   LIMITS";U(M);"TO";U(M+1)
640 NEXT M
650 PRINT
660 PRINT
670 FOR M=1 TO N3
680 REM FETCH NO. SEGMENTS PER LINE, THEN BY PAIRS,
690 REM AREA NO. AND RIGHMOST PRINT POSITION FOR EACH.
700 GET 1: P1
710 FOR N=1 TO P1
720 GET 1: A(N),B(N)
730 NEXT N
740 I1=0
750 FOR N=1 TO P1
760 I2=I1+1
770 I1=B(N)
780 K=A(N)
790 K=D(K)
800 FOR J=I2 TO I1
810 E$(J)=C$(K)
820 NEXT J
830 NEXT N
840 IF I1>= L THEN 890
850 I2=I1+1
860 FOR J=I2 TO L
870 E$(J)=C$(N4)
880 NEXT J
890 L1=L-1
900 FOR N=1 TO L1
910 PRINT E$(N);
920 NEXT N
930 PRINT E$(L)
940 NEXT M
950 REM END COMPOSE AND PRINT SEQUENCE.
960 CLOSE 1
970 FOR N=1 TO 5
980 PRINT
990 NEXT N
1000 PRINT "DO YOU WANT TO MAKE ANOTHER MAP (YES OR NO)?"
1010 INPUT A$
1020 IF A$='YES' THEN 210
1030 STOP
1040 END
```

Figure 1. Program CMAPS Listing.

```
PROGRAM CMAPS MAKES CHOROPLETH MAPS.  SEE **CMAPSDOC.

ENTER THE MAP TITLE (UP TO 18 CHARS).
?PCT 1&2 CANADIAN
ENTER THE NO. OF CLASSES, AREAS, SCAN LINES,
AND MAP WIDTH IN COLUMNS.
?3,10,50,45
ENTER THE LOWER LIMIT FOR THE FIRST CLASS AND,
IN ORDER, THE UPPER LIMIT FOR EACH CLASS.
?0,10,20,35

ENTER A PRINT SYMBOL FOR EACH CLASS, ONE AT A TIME,
IN ORDER.  THE SYMBOLS '.' AND 'R' ARE RESERVED.
?0
?X
?#
ENTER THE DATA TO BE MAPPED, ONE VALUE FOR EACH
MAP AREA.  THE ORDER MUST MATCH SCAN DECK NUMBERING.
?30.7,9.6,7.3,12.8,12.0,10.5,12.4,8.7,7.7,17.1

WHAT IS THE NAME OF YOUR SCAN-LINE FILE?
?NHSCAN
```

Figure 2. Sample CMAPS Run.

```
PCT 1&2 CANADIAN

CLASS 1    SYMBOL 0   LIMITS 0    TO 10
CLASS 2    SYMBOL X   LIMITS 10   TO 20
CLASS 3    SYMBOL #   LIMITS 20   TO 35.0003

..........................................
..........................................
.............................##...#.......
...........................########.......
..........................########........
.........................#########........
.........................#########........
.........................#########........
.........................#########........
.........................#########........
........................##########........
........................##########........
........................##########........
........................##########........
........................##########........
........................##########........
........................##########........
.....................00################....
...................000000################..
.................0000000000#############0....
...............000000000000###000000.......
..............0000000000000#0000000.......
..............0000000000000000000000.......
..............0000000000000000000000.......
..............0000000000000000
```

```
LINE NO. 1          1     11    45
LINE NO. 2          1     11    45
LINE NO. 3          5     11    30    1     32    11    35    1     36
  11    45
LINE NO. 4          3     11    28    1     36    11    45
LINE NO. 5          3     11    28    1     36    11    45
LINE NO. 6          3     11    27    1     36    11    45
LINE NO. 7          3     11    27    1     36    11    45
LINE NO. 8          3     11    26    1     36    11    45
LINE NO. 9          3     11    26    1     36    11    45
LINE NO. 10         3     11    26    1     37    11    45
LINE NO. 11         3     11    25    1     37    11    45
LINE NO. 12         3     11    24    1     37    11    45
LINE NO. 13         3     11    24    1     37    11    45
LINE NO. 14         3     11    25    1     37    11    45
LINE NO. 15         3     11    26    1     37    11    45
LINE NO. 16         3     11    25    1     37    11    45
LINE NO. 17         3     11    24    1     37    11    45
LINE NO. 18         3     11    23    1     37    11    45
LINE NO. 19         4     11    20    2     22    1     37    11    45
LINE NO. 20         4     11    17    2     23    1     37    11    45
LINE NO. 21         5     11    15    2     25    1     37    3     38
  11    45
LINE NO. 22         5     11    15    2     28    1     32    3     38
  11    45
LINE NO. 23         6     11    15    2     29    3     30    1     31
  2     38    11    45
LINE NO. 24         4     11    15    2     29    3     38    11    45
LINE NO. 25         4     11    14    2     30    3     38    11    45
LINE NO. 26         4     11    14    2     30    3     38    11    45
LINE NO. 27         4     11    13    2     30    3     38    11    45
LINE NO. 28         4     11    12    2     25    3     35    11    45
LINE NO. 29         4     11    12    2     26    3     35    11    45
LINE NO. 30         4     11    11    2     26    3     38    11    45
LINE NO. 31         5     11    9     2     26    4     28    3     38
  11    45
LINE NO. 32         5     11    9     2     24    4     30    3     38
  11    45
LINE NO. 33         5     11    8     2     23    4     31    3     38
  11    45
LINE NO. 34         8     11    7     5     13    2     18    6     22
  4     34    7     35    3     39    11    45
LINE NO. 35         6     11    7     5     17    6     23    4     33
  2     30    11    45
LINE NO. 36         6     11    7     5     15    6     27    4     34
  2     39    11    45
LINE NO. 37         6     11    7     5     14    6     29    4     34
  2     39    11    45
LINE NO. 38         6     11    7     5     14    6     31    4     33
  2     40    11    45
LINE NO. 39         5     11    6     5     14    6     33    7     41
  11    45
LINE NO. 40         6     11    6     5     15    6     31    8     34
  2     42    11    45
LINE NO. 41         9     11    6     5     7     9     12    5     15
  10    18    6     31    8     36    7     42    11    45
LINE NO. 42         8     11    6     9     15    10    23    6     30
  8     36    7     41    8     43    11    45
LINE NO. 43         6     11    6     9     15    10    27    6     29
  8     44    11    45
LINE NO. 44         5     11    5     9     15    10    29    8     44
  11    45
LINE NO. 45         5     11    4     9     15    10    28    8     43
  11    45
LINE NO. 46         5     11    3     9     16    10    28    8     40
  11    45
LINE NO. 47         5     11    4     9     15    10    29    8     35
  11    45
LINE NO. 48         5     11    5     9     17    10    31    8     34
  11    45
LINE NO. 49         4     11    6     9     18    10    32    11    45
LINE NO. 50         1     11    45
```

Figure 3. NHSCAN File.

```
100 REM PROGRAM TO PREPARE SCAN-LINE FILES.
110 REM JAMES M. GERNY, APRIL 1975.
120 PRINT "PROGRAM TO PREPARE SCAN-LINE FILES.  SEE **CMAPSDOC."
130 PRINT
140 DIM B(50)
150 PRINT "FILE NAME?"
160 INPUT A$
170 OPEN 1,A$,OUTPUT
180 PRINT
190 K=0
200 PRINT "NEW BOW.  ENTER NO. SECS.  ZERO TO STOP."
210 INPUT A
220 IF A=0 THEN 300
230 K=K+1
240 PUT 1: A
250 M=A*2
260 PRINT "ENTER SEGMENT PAIRS: AREA NO., RIGHT PRINT POS."
270 MAT INPUT B(M)
280 MAT PUT 1: B
290 GO TO 200
300 CLOSE 1
310 OPEN 1,A$,INPUT
320 FOR M=1 TO K
330 GET 1: A
340 PRINT A;
350 M=A*2
360 MAT GET 1: B(M)
370 MAT PRINT B
380 NEXT M
390 STOP
400 END
```

Figure 4. Program SCANMAKE.

Dogbite

Randall Lockwood, Ph.D.

A Public Health Simulation

"Dog Bites Man" is not news, largely because it is very common. At least 2% of the U.S. population is bitten by dogs each year, seriously enough to require medical attention. The victims are usually children between the ages of 5 and 9. As large dogs are becoming more popular, the injuries are becoming more serious.

Among adults, certain occupations are associated with a disproportionate number of bites. Letter carriers and meter readers are bitten about 15 times as often as other adults. Unfortunately, society has tended to take these problems lightly. You can even buy mailman-shaped biscuits for your dog! However, public health officials and victims are starting to recognize the "epidemic" proportions of the problem.

DOG BITE grew out of a belief that simulations based on real-world situations can be both entertaining and useful. (See "Jury Selection" and "Real World Games", **Creative Computing,** volume 4, number 5). DOG BITE is based on a series of studies conducted with the cooperation of the St. Louis branch of the U.S. Postal Service. In these studies, Dr. Alan Beck and I reviewed all accident reports for dog bites to letter carriers in St. Louis during 1971-1972. These were compared to reports of non-dog related accidents. In addition, I spent a year working the routes with letter carriers, noting the incidence of dogs, the interactions that occurred, the areas of greatest risk and the effectiveness of different responses to dangerous situations. I have tried to include both my statistical findings and my intuitive knowledge of dog-human encounters in DOG BITE to create a fun and challenging simulation.

Randall Lockwood, Ph.D., Dept. of Psychology, State University of New York, Stony Brook, NY 11794.

Background

There appear to be four main factors that affect the dog-bite history of a letter carrier. The first is the nature of the assignment. Carriers who visit the same houses each day have a chance to learn where the dogs are, particularly the nasty ones. At the same time, the dogs have a chance to get used to a regular intruder. However, mail is delivered 6 days a week and most carriers work only 5. On these odd days a new person shows up, and is thus a more likely target. Other carriers routinely fill in for those who are absent or ill, and thus have the same problems.

Another important variable is the nature of the area being served. Urban areas, particularly apartment buildings, are safe as far as dog bite is concerned. The dogs are usually small and inside and many deliveries can be made at one time. Suburban areas with one and two family homes present greater dangers. Dogs are likely to be running loose or to get loose during a delivery. Stray dogs (i.e., those with no owners) are not much of a problem in DOG BITE or in real life. The greatest problems are caused by straying pets, or animals that get out when a customer opens the door to receive mail.

Once a letter carrier faces a confrontation with a dog, the third factor becomes important: the ability to "read" the animal's intentions and react accordingly. Even friendly animals should not be encouraged to approach (and this is **not** an option in DOG BITE), since their reaction to the uniform and apparatus of the letter carrier may be unpredictable. We found several carriers who were bitten by their own dogs after they put on their uniforms.

The letter carrier has the options of attempting delivery, withholding delivery, running away, confronting the animal or using a chemical repellent. Most carriers avoid using a spray since they want to avoid possible injury to dogs, and unpleasant reactions from dog-owners. You will face the same kinds of problems in using chemical sprays in DOG BITE.

The letter carriers' success in avoiding a bite is also related to a fourth factor . . . LUCK. Even careless people may get away unharmed and even the most cautious may be bitten.

DOG BITE provides a good opportunity to familiarize yourself with the many factors that interact in a public health simulation. Hopefully, playing DOG BITE may even help you avoid becoming a statistic in our next program!

About The Program

DOG BITE is written, appropriately, in "PET" Basic. However, the version presented here does not make use of any of the special PET graphics or cursor control commands, so it is easily adapted to all Basic dialects. The only peculiarities are line 15, which takes the place of a "RANDOMIZE" instruction, and the use of the GET command instead of INPUT or INKEY$. I suggest that you add the "CLEAR/HOME" instruction for your system before lines 10, 100, 455, 540 and 1505. This will make things a bit neater. PET owners should have the warning messages print in reverse video to add to the drama.

In writing a public health simulation, one is often in the situation of having numerator data without denominators. For example, we know how many biting dogs are poodles, but we have to rely on other sources to try to estimate the proportion of poodles that bite. To keep DOG BITE true-to-life, while realizing that we cannot perfectly predict dangerous situations, I have made some compromises. I have aimed for an average of 3 dangerous encounters per game. This is high, but a more realistic game would be dull. The *relative* rates of the various events (dog loose, dog is barking, etc.) are consistent with the actual data.

The general design of DOG BITE is to assign certain basic risk factors and then increase or decrease the risks according to the situations that arise and the choices made by the player. If the risk factors reach a critical level and the situation is right — a bite will occur. More specifically:

lines 235-540:
assign basic risk (BR) and adjust the risk factor, RF according to the assignment

lines 545-820:
select the type of delivery and evaluate the risks associated with that type of delivery according to the carrier's assignment

lines 830-995:
set up a confrontation and adjust RF according to the dog's temperament (determined at 4000-4300) and player's choices.

line 1000:
determines if RF is great enough for a bite

lines 1010-1210:
determine the nature of bite injury

lines 1500-1550:
report on performance in the simulation. □

References

Beck, A.M., Loring, H. and Lockwood, R. 1975. The ecology of dog bite injury in St. Louis, Mo., *Public Health Reports*, 90(3):262-267.

Lockwood, R. and Beck, A.M. 1975. Dog bites among letter carriers in St. Louis, *Public Health Reports*, 90(3):267-269.

A cassette with a PET BASIC version of DOG BITE, along with copies of the original research reports and documentation are available from the author for $2.00.

```
1 REM R.LOCKWOOD/DEPT.OF PSYCH./SUNY,STONY BROOK,11794
5 PRINTSPC(14);"DOG BITE
15 ZZ=RND(-TI):REM RANDOMIZE
20 PRINT:PRINT"THIS SIMULATION IS BASED ON ACTUAL
25 PRINT"DOG ATTACK DATA COLLECTED BY R .LOCKWOOD
30 PRINT"AND A. BECK IN ST. LOUIS,MO.
40 PRINT:PRINT" THE LIKELIHOOD OF AN ENCOUNTER
45 PRINT"WITH A DOG HAS BEEN MADE HIGHER
50 PRINT"IN THIS SIMULATION TO MAKE THINGS
55 PRINT"MORE INTERESTING.
60 PRINT:PRINT" THE OUTCOME OF AN ENCOUNTER WILL
65 PRINT"DEPEND ON YOUR DECISIONS AS WELL AS
70 PRINT"ON ENVIRONMENTAL AND CHANCE FACTORS.
80 PRINT:PRINT" ARE YOU READY TO BEGIN?
82 GET A$:IF A$="" THEN 82
90 IFA$<>"Y"THEN82
95 GOTO110
100 PRINT:PRINT:PRINT"   YOU ARE A LETTER CARRIER IN THE
110 PRINT:PRINT:PRINT"   YOU ARE A LETTER CARRIER IN THE
120 PRINT"CITY OF GNAWSHVILLE,WHICH HAS A DOG
125 PRINT"BITE PROBLEM.
130 PRINT:PRINT"   YOU ARE TO ATTEMPT TO MAKE 10
135 PRINT"DELIVERIES ON YOUR ROUTE. IF YOU DO
140 PRINT"THIS WITHOUT BEING BITTEN OR ALIEN-
145 PRINT"ATING YOUR CUSTOMERS,YOU WIN!":PRINT:PRINT
148 RA=0:H=0:N=0:W=0:ND=0:B=0
150 PRINT"CHECKING MY RECORDS, I SEE THAT YOU
155 PRINT"HAVE BEEN BITTEN ";PB;" TIME";
160 IF PB=1THEN PRINT" BEFORE.":GOTO180
165 PRINT"S BEFORE.
170 IF PB=0 THEN 190
180 PRINT"YOU'D BETTER BE MORE CAREFUL!!" GOTO200
190 PRINT"YOU'RE EITHER NEW TO THIS GAME OR
195 PRINT"SIMPLY LUCKY...WATCH OUT!
200 PRINT:PRINT:PRINT"DO YOU WANT TO (1) CHOOSE AN
205 PRINT"ASSIGNMENT OR (2) HAVE ME ASSIGN
210 PRINT"YOU? (I'LL TRY TO BE FAIR).TYPE
215 PRINT"THE APPROPRIATE NUMBER.
220 GETC$:IFC$=""THEN220
221 C=VAL(C$):IFC>2THEN220
222 IFC=0THEN220
230 IF C=1 THEN 280
234 PRINT:PRINT:PRINT
235 PRINT:PRINT"O.K. I'LL GIVE YOU AN ASSIGNMENT
240 GOSUB 2000
245 PRINT"FIRST I'LL ASSIGN A SCHEDULE
246 PRINT
250 IF R<=70 THEN S=1:GOTO 310
255 IF R<=85 THEN S=2:GOTO 310
260 S=3:GOTO 310
280 PRINT:PRINT" FIRST PICK AN ASSIGNMENT:":PRINT
285 PRINT"     (1) REGULAR CARRIER
290 PRINT"     (2) UTILITY CARRIER
300 PRINT"     (3) SUBSTITUTE CARRIER■
305 PRINT"YOUR CHOICE ? (ENTER NUMBER):
306 GETM$:IFM$=""THEN306
307 S=VAL(M$):IFS=0THEN306
308 IFS>3THEN306
310 IFS=1THENS$="REGULAR CARRIER
315 IFS=2THENS$="UTILITY CARRIER
320 IFS=3THENS$="SUBSTITUTE CARRIER
321 PRINT:PRINT" YOU ARE A "S$
330 GOSUB 2000
340 BR=INT(25*RND(1)+1)
360 IFS=1THENBR=BR+10
370 IFS=2THENBR=BR+30
380 IFS=3THENBR=BR+30
400 PRINT:PRINT"HERE IS SOMETHING YOU SHOULD KNOW:
405 ON S GOTO 410,420,440
410 PRINTS$;"S ARE KNOWN TO LOCAL"
415 PRINT"DOGS AND THUS GET FEWER BITES.":GOTO455
420 PRINTS$;"S COVER VARIOUS ROUTES
425 PRINT"EACH DAY AND DON'T GET TO KNOW THE
430 PRINT"LOCAL DOGS.
435 GOTO455
440 PRINTS$;"S FILL IN FOR
445 PRINT"REGULARS AND ARE STRANGERS
450 PRINT"TO LOCAL DOGS.
455 PRINT:PRINT"HIT ANY KEY TO CONTINUE
460 GETK$:IFK$=""THEN460
480 PRINT:PRINT:PRINT"    GNAWSHVILLE HAS FOUR NEIGHBOR
    HOODS:":PRINT
482 PRINT"  (1)PUNY ARMS-LUXURY APTS.":PRINT
484 PRINT"  (2)SUPERCILIOUS MANOR-UPPER
485 PRINT"     INCOME SINGLE FAMILY HOUSES":PRINT
487 PRINT"  (3)TACKYTOWN-MID INCOME 1 AND 2
488 PRINT"     FAMILY HOUSES
490 PRINT:PRINT"  (4)RATHAVEN-LOW INCOME 2 FAMILY"
491 PRINT"     HOUSES":PRINT
500 IFC=1THEN522
502 PRINT"I'LL ASSIGN YOU A NEIGHBORHOOD"
505 GOSUB 2000
510 IFR>=90 THEN A=1:GOTO525
515 IF R>=70 THEN A=2:GOTO 525
520 IF R>=30 THEN A=3:GOTO 525
521 A=4:GOTO 525
522 PRINT"PICK A NEIGHBORHOOD (1-4):
523 GETG$:IFG$=""THEN523
524 A=VAL(G$):IFA=0THEN523
525 IFA>4THEN523
526 IFA=1THENT$="PUNY ARMS"
527 IFA=2THENT$="SUPERCILIOUS MANOR"
528 IFA=3THENT$="TACKY TOWN"
529 IFA=4THENT$="RAT HAVEN"
530 PRINT:PRINT"YOU'LL BE GOING TO ";T$
535 PRINT:PRINT"HIT ANY KEY TO BEGIN
536 GETX$:IFX$=""THEN536
540 RA=0:RF=BR:OP=0:RA=0:N=N+1:IF N=11 THEN 1500
544 IFN>1THENGOSUB3000
545 PRINT:PRINT:PRINTSPC(8);"DELIVERY # ";N;:PRINT:PRINT:
    GOSUB2000
550 IF R<=60 THEN D$="REGULAR MAIL":D=1:GOTO 570
555 IF R<=80 THEN D$="A PACKAGE":D=2 GOTO570
560 IF R<=90 THEN D$="A SPECIAL DELIVERY":D=2 GOTO 570
565 D$="A POSTAGE DUE LETTER":D=2:GOTO 570
570 PRINT"YOUR DELIVERY IS ";D$
575 IF D=1 THEN 580
577 PRINT"THE CUSTOMER WILL HAVE TO OPEN"
578 PRINT"THE DOOR TO GET IT.":PRINT:PRINT
579 FOR X=1 TO 500:NEXT X
580 GOSUB 2000
585 IF A=1 AND R>=60 THEN 640
590 IF A=2 AND R>=40 THEN 640
600 IF A=3 AND R>=40 THEN 640
610 IF A=4 AND R>=65 THEN 640
615 PRINT"THERE DON'T SEEM TO BE ANY DOGS":PRINT
620 GOSUB2000
630 IF R>8 THEN 1025
635 PRINT"WAIT!...HERE COMES A STRAY!":GOTO850
640 PRINT"** DOG ALERT **"
650 GOSUB 2000
660 IF A=1 AND R>=65 THEN 720
670 IF A=2 AND R>=50 THEN 720
680 IF A=3 AND R>=60 THEN 720
690 IF A=4 AND R>=40 THEN 720
700 IF R<=10 THEN PRINT "THE DOG IS SECURELY CONFINED" GOTO 1025
710 PRINT"THE DOG IS INSIDE":F=0:GOTO 750
720 F=1:RF=RF+50:PRINT:PRINT"**THE DOG IS LOOSE**"
750 PRINT:PRINT"WHAT DO YOU WANT TO DO?
760 PRINT:PRINT"     (1) ATTEMPT DELIVERY"
770 PRINT"     (2) WITHHOLD DELIVERY":PRINT
772 GETQ$:IFQ$=""THEN772
773 Q=VAL(Q$)
774 IFQ<1ORQ>2THEN772
776 IF Q=1 THEN 800
780 PRINT"O.K., DELIVERY WITHHELD.
781 PRINT:PRINT"WE'LL GO ON TO THE NEXT HOUSE..." PRINT
782 W=W+1:GOTO 540
800 IF F=1 THEN 850
810 GOSUB 2000
815 IF R<=60 AND D=2 THEN 830
820 GOTO1025
830 PRINT:PRINT"** THE OWNER LET THE DOG OUT **":PRINT
850 GOSUB 4000                                    RF=RF+60
```

```
870 PRINT"** IT'S COMING TOWARDS YOU"
875 PRINT"WHAT DO YOU WANT TO DO?
880 PRINT:PRINT"    (1)RUN AWAY
885 PRINT"    (2)USE CHEMICAL SPRAY
890 PRINT"    (3)STAND STILL
900 PRINT"    (4)WALK AWAY":PRINT
902 GETZ$:IFZ$=""THEN902
903 Z=VAL(Z$):IFZ=0THEN902:IFZ>4THEN902
910 ON Z GOTO930,935,970,990
930 RF=RF+20:ND=ND+1 RA=1:GOTO1000
935 IF RND(1)>.4THEN950
940 PRINT"YOU CAN'T GET IT OUT !!!..."
945 RF=RF+5:GOTO 1000
950 PRINT"PSSSSSSTTTTTTTT.....":PRINT
952 IF DF<=10THENPRINT"YOU SPRAYED A HARMLESS DOG!!"
955 IFDF>25THENPRINT"THAT WAS CLOSE !"
960 RF=RF-30:GOSUB2000:IFR<50ANDOP=0THEN1000
965 PRINT"THE CUSTOMER SAW THAT!!!":H=H+1
967 IFCP<>0THENPRINT"YOU SHOULDN'T USE HALT AROUND KIDS!"
968 GOTO1000
970 IFRF<=100THENPRINT"GOOD JUDGEMENT!"
975 RF=RF-20:GOTO1000
990 IF RF<=1000THENPRINT"GOOD CHOICE.":PRINT
995 RF=RF-20:ND=ND+1:RA=1
1000 IF RF>100 THEN 1060
1010 PRINTSPC(6);"**** BITE AVOIDED ****":PRINT
1020 IF RA=1 THEN 1040
1025 PRINT"DELIVERY COMPLETED";
1035 GOTO 540
1040 PRINT"BUT YOU FAILED TO MAKE THE DELIVERY."
1050 GOTO540
1060 PRINT:PRINT"!!!! A BITE !!!!":ND=ND+1
1065 GOSUB 2000:B=1:PB=PB+1
1067 PRINT:PRINT"HE GOT YOUR ";
1070 IFR>90THEN1135
1080 IFRND(1)<.5THENPRINT" RIGHT ";:GOTO1115
1090 PRINT" LEFT ";
1115 IF R<=35 THEN PRINT"LEG.":GOTO 1200
1120 IF R<=70 THEN PRINT"FOOT.":GOTO 1200
1125 IF R<=80 THEN PRINT"ARM.":GOTO 1200
1130 IF R<=90 THEN PRINT"HAND.":GOTO 1200
1135 PRINT "BODY.": GOTO 1200
1200 PRINT"YOU WILL NEED TREATMENT."
1205 PRINT"YOU'RE OUT OF THE GAME!"
1210 N=N+1:GOTO1500
1490 FORZZ=1 TO 1000:NEXT
1500 PRINT:PRINT:PRINT
1505 PRINT"********** SUMMARY **********":PRINT:PRINT
1510 PRINT"YOU COMPLETED ";N-W-ND-1;" DELIVERIES."
1515 IF B=0 THEN 1525
1520 PRINT"AND WERE BITTEN.":GOTO 1530
1525 PRINT"AND WERE NOT BITTEN.":PRINT
1530 PRINT"YOU ALIENATED ";H+W+ND;"CUSTOMER(S)"
1550 PRINT"BY USING HALT OR FAILING TO DELIVER.":PRINT:PRINT
1560 PRINT"DO YOU WANT TO TRY AGAIN?
1570 GET Q$:IF Q$="" THEN 1570
1580 IFQ$="N"THEN9999
1590 GOTO 148
1995 GOTO9999
2000 R=INT(100*RND(1)+1):RETURN
3000 FORDD=1TO5000:NEXT:RETURN
4000 REM DOG FACTORS
4005 DF=0
4010 IFR<=40THENT$="GERMAN SHEPHERD":DF=DF+10:GOTO4070
4020 IFR<=55THENT$="MUTT":DF=DF+5:GOTO4070
4030 IFR<=65THENT$="POODLE":DF=DF+10:GOTO4070
4040 IFR<=85THENT$="BIG DOG":DF=DF+10:GOTO4070
4050 IFR<=95THENT$="LITTLE DOG":GOTO4070
4060 T$="DOBERMAN"
4070 PRINT"IT'S A ";T$;"."
4075 REM BARK
4080 GOSUB2000
4085 IFR<=30THENPRINT"IT'S NOT BARKING":DF=DF-10:GOTO4100
4090 IFR<=80THENPRINT"IT'S BARKING AT YOU!":
     DF=DF+INT(RND(1)*6):GOTO4100
4095 PRINT"IT'S BARKING AND GROWLING!":DF=DF+10
4100 REM TAIL
4105 GOSUB2000
4110 IFR<33THEN PRINT"IT'S TAIL IS DOWN.":DF=DF-10:GOTO4200
4120 IFR<=66THENPRINT"IT'S TAIL IS UP AND WAGGING SLOWLY.":
     DF=DF+5:GOTO4200
4130 PRINT"IT'S TAIL IS UP AND WAGGING FAST!":DF=DF+5
4200 REM OWNER PRESENT
4205 GOSUB2000:OP=0
4210 IFR<=60THEN4300
4220 IFR<=75THENPRINT"THE OWNER IS OUTSIDE.":OP=1:DF=DF+10:
     GOTO4300
4230 PRINT"THERE IS A CHILD NEARBY.":DF=DF+10:OP=1:CP=1
4300 RETURN
9999 PRINT"O.K., THANK'S FOR PLAYING DOGBITE!":END
```

SAMPLE RUN

```
RUN

   THIS SIMULATION IS BASED ON ACTUAL
DOG ATTACK DATA COLLECTED BY R. LOCKWOOD
AND A. BECK IN ST. LOUIS, MO.

 THE LIKELIHOOD OF AN ENCOUNTER WITH
A DOG HAS BEEN MADE HIGHER IN THIS
SIMULATION TO MAKE THINGS MORE
EXCITING.

   THE OUTCOME OF AN ENCOUNTER
WILL DEPEND ON YOUR DECISIONS AS WELL
AS ON ENVIRONMENTAL FACTORS AND CHANCE.

HIT ANY KEY TO BEGIN

Y
  YOU ARE A LETTER CARRIER IN THE
CITY OF GNAWSHVILLE, WHICH HAS A DOG
BITE PROBLEM.

   YOU ARE TO ATTEMPT TO MAKE 1Ø
DELIVERIES ON YOUR ROUTE.  IF YOU
DO THIS WITHOUT BEING BITTEN OR ALIEN -
ATING YOUR CUSTOMERS... YOU WIN!

CHECKING MY RECORDS I SEE THAT
YOU HAVE BEEN BITTEN  Ø TIMES
BEFORE.
YOU'RE EITHER NEW TO THIS GAME OR
SIMPLY LUCKY...WATCH OUT!!

DO YOU WANT TO (1) CHOOSE AN
ASSIGNMENT OR (2) HAVE ME ASSIGN YOU?
(I'LL TRY TO BE FAIR)
TYPE THE APPROPRIATE NUMBER
2

O.K., I'LL GIVE YOU AN ASSIGNMENT
FIRST I'LL ASSIGN YOU A SCHEDULE

YOU ARE A SUBSTITUTE CARRIER

HERE IS SOMETHING YOU SHOULD KNOW:

SUBSTITUTE CARRIERS FILL IN FOR REGULARS
AND ARE CONSIDERED TO BE STRANGERS
BY THE LOCAL DOGS.
HIT ANY KEY TO CONTINUE

GNAWSHVILLE HAS FOUR NEIGHBORHOODS

  (1)PUNY ARMS-LUXURY APTS.

  (2)SUPERCILIOUS MANOR-UPPER INCOME
     SINGLE FAMILY HOUSES

  (3)TACKYTOWN-MID INCOME 1 AND
     2 FAMILY HOUSES

  (4)RATHAVEN-2 FAMILYHOUSES

I'LL ASSIGN YOU A NEIGHBORHOOD.

YOU'LL BE GOING TO TACKYTOWN

HIT ANY KEY TO BEGIN

          DELIVERY # 1

YOUR DELIVERY IS REGULAR MAIL
THERE DON'T SEEM TO BE ANY DOGS
DELIVERY COMPLETED

          DELIVERY # 2

YOUR DELIVERY IS A PACKAGE
THE CUSTOMER WILL HAVE TO OPEN
THE DOOR TO GET IT.

***** DOG ALERT *****

THE DOG IS INSIDE.

WHAT DO YOU WANT TO DO?

   (1) ATTEMPT DELIVERY
   (2) WITHHOLD DELIVERY


2
O.K. ,DELIVERY WITHHELD.
WE'LL GO ON TO THE NEXT HOUSE.

          DELIVERY # 3

YOUR DELIVERY IS REGULAR MAIL

***** DOG ALERT *****

THE DOG IS SECURELY CONFINED.
DELIVERY COMPLETED

          DELIVERY # 4

YOUR DELIVERY IS A POSTAGE DUE LETTER
THE CUSTOMER WILL HAVE TO OPEN
THE DOOR TO GET IT

***** DOG ALERT *****

THE DOG IS INSIDE.

WHAT DO YOU WANT TO DO?

   (1) ATTEMPT DELIVERY
   (2) WITHHOLD DELIVERY
1

!!!! THE OWNER LET THE DOG OUT !!!!

****  IT'S COMING TOWARDS YOU ****

IT'S A GERMAN SHEPHERD

IT'S BARKING
IT'S TAIL IS UP AND WAGGING SLOWLY
THE OWNER IS PRESENT.

WHAT DO YOU WANT TO DO?

   (1) RUN AWAY
   (2) USE CHEMICAL SPRAY
   (3) STAND STILL
   (4) WALK AWAY SLOWLY

1

##### A BITE #####

THE DOG GOT YOUR RIGHT HAND
YOU WILL NEED TREATMENT
YOUR'RE OUT OF THE GAME

---------- SUMMARY ----------

YOU COMPLETED 2 DELIVERIES
AND WERE BITTEN!
YOU ALIENATED 2 CUSTOMER(S)
BY USING THE SPRAY OR FAILING TO DELIVER

DO YOU WANT TO TRY AGAIN?

N
 THANKS FOR PLAYING DOGBITE!!!!!
```

Futurists will Explore Possible Events Of 1980s

The 1980s will be the focus of a major conference in Toronto, Canada in July.

Called the First Global Conference on the Future, the meeting will convene some 4,000 scholars, decision-makers and other futurists from around the world for a week of deliberations on the theme "Through the 80s: Thinking Globally, Acting Locally."

The conference is being organized by the World Future Society and the Canadian Futures Society.

Top thinkers and doers will address the conference, including: Italian industiralist Aurelio Peccei, founder of the Club of Rome; Filipino official Rafael Salas, now head of the United Nations' population activities; German author Robert Jungk; Indian urban planner Rashmi Mayur; French social scientist Bertrand de Jouvenel; Argentine economist Andre van Dam; Canadian communications theorist Marshall McLuhan; Swiss economist Bruno Fritsch; and United States' policy analyst Herman Kahn; author-diplomat Harlan Cleveland; public opinion analyst Florence Skelly; environmentalist Lester R. Brown and more.

For information, write: World Future Society, 4916 St. Elmo Avenue, Washington, DC 20014, U.S.A.

Map of the 1980s: The World Future Society recently published this "Roadmap of the 1980s." Society president Edward Cornish explains: "The map is humorous but makes an important point. The decade we have just entered contains enormous possibilities and we face a great many important decisions that will determine whether we have a good or bad future."

Chapter Two

Ecology and Biology

Seeing is Believing, But Simulating is Convincing

Walter Koetke

Simulation represents one of the more promising areas in which the computer can be used to provide pertinent data on the options facing our society. Many computer related texts, however, pursue the subject no further than Buffon's needle, playing craps or dealing cards. While these are indeed valid examples of simulation, they are not sufficient because: they do not clearly demonstrate a potential connection between simulation results and human decision making; they seem to associate simulation with theoretical problems of mathematics as opposed to real problems of society; and they are too simple to give a feeling for the true complexity of societal simulations. More elaborate simulations are available (such as the variety of material from the Huntington Project), but these are intended for students to execute rather than write. The following two problems are neither outstanding problems of society nor outstanding problems of mathematics. They are offered because they provide the student with the opportunity to write simulations that go just a little further than the standard examples.

Problem 1: Horse Ranch. Suppose you are a rancher and own 200 horses. All of your stock is healthy, and that's very important because you expect a buyer to arrive unannounced sometime in the next 10 days. The buyer is looking for 160 healthy horses. You know from experience that if he finds fewer than 160 healthy animals, he won't buy any at all. Clearly your finances could not absorb the complete loss of the entire sale.

Just as you're ready to celebrate the pending sale, you learn that one of your horses has "day cough", a terrible sounding disease of short duration. Day cough lasts exactly one day and immunity to re-exposure results. You know that each sick horse will contact five other horses each day, and each contact has 0.6 chance of transmitting the sickness if the contacted horse is healthy and not immune. Although your situation appears bleak, do you really have such a serious problem? Which of the following three alternatives is your best course of action and why is it best?

a) Stop worrying. There's really very little chance that fewer than 160 horses will be healthy during the next 10 days.

b) Solicit the expensive support of modern medical science. "Cough shots" are available that are supposed to provide instant immunity when given to a healthy horse. However, only 13 horses per day can be innoculated by the local veterinarian, and neither he nor you can tell a healthy horse from a horse already immune. Thus you may innoculate an already immune horse. The only problem in doing this is that one of the cough shots is wasted. Actually, the cough shots have a 90% chance of providing instant immunity and a 10% chance of instantly giving day cough to a healthy horse. For all his expertise and advice, the veterinarian will charge you one healthy horse for each day of service he provides.

c) By calling right away, you can probably convince the buyer to delay his unannounced visit for 10 days. He will still come unannounced, but during the period 10 to 20 days from now rather than during the next 10 days.

Problem 2: Fish Pond. As part of a conservation and ecology project, a group of biology students has designed the following controlled experiment: A small pond is polluted with several common types of waste material. Exactly 100 male fish are then introduced to the previously fish free pond. Each day the students carefully net exactly 10 fish. All 10 fish are caught simultaneously and at the same time each day. The netted fish are examined for signs of gill disease, their tails dyed, and they are returned to the pond. The dye used is harmless and completely disappears after 13 full days in the water. If a netted fish is already dyed, it is dyed again so that it too will remain dyed for the next 13 days. The experiment continues until each of the fish netted on any one day all have dye on their tails.

How many days should the students allow to permit "a reasonable chance" for successful completion of the experiment?

A complete solution to this problem, as in many problems of society for which computer simulations might be useful, requires clarification of some human values. The definition of "a reasonable chance" is a personal one, and as such it is based on a wide variety of factors. Discussing the point alone makes this a very worthwhile problem in a classroom setting.

A closely related simulation problem can be described by not stating the number of fish in the pond, but instead specifying the number of days that pass before the 10 fish netted all have dye on their tails. The question then becomes "How many fish were originally in the pond?" For instance, if the experiment described was completed in 32 days, how many fish were originally placed in the pond?

Problem 3: Superspy. This one is included just for fun. The problem is really another disguise of the two dimensional random walk.

Each night IBF, the superspy, leaves his daytime refuge and emerges from a secret manhole cover in the center of a city. Being an exceptionally tricky spy, IBF is likely to sneak forward, backward, left or right at every intersection. If IBF happens to accidentally stray 8 blocks from the manhole (in any direction) he is captured by the arch enemies of society, the TUVEFOUT. If IBF happens to return to his manhole, he is safe for another night. What is the probability that IBF will return safely and thus complete a successful mission? What is the average number of blocks traveled during a mission?

Ecology: A Simulation Model for Small Populations of Animals

James W. Grier

The impact of human activities on the populations of other species has been receiving increased interest and concern. The issue of the snail darter, a fish endangered by the proposed Tellico Dam, is probably the best known; but it is only one of numerous examples. Other cases involve the black-footed ferret, the timber wolf, some species of prairie chickens, the peregrine falcon, and many others; even our nation's symbol, the bald eagle.

Wildlife ecologists are developing new techniques for managing and attempting to save such species, by transplanting eggs, or young, or whole populations to different locations, or by breeding the species in captivity for reintroduction to the wild. Ecologists are also attempting to isolate and correct the most critical factors limiting the populations in the wild. A recent reference for work on birds is *Endangered Birds,* edited by S. A. Temple (Univ. of Wisc. Press, 1977).

One of the biggest problems in working with small populations, however, is the large part played by chance events — which can lead to several different outcomes. If one has only four animals — two females and two males, for example — and the expected mortality rate is 50 percent, the whole future of the population rests on *which* 50 percent dies. If by chance it is 1 female and 1 male, there is still a pair left which may reproduce and increase the population severalfold. If, however, the two females die, only the males remain and the population is doomed.

Chance can enter the picture at several points. Not only which individuals die, but how many young each female produces and the sex of those young, are chance determinations which affect the fate of the population.

The problem with most population models is that they are *deterministic,* that is, they are determined by a fixed set of values with no room for chance and with only one outcome. A classic example of such a model is the age-specific life table (for a reference, see C. J. Krebs. *Ecology.* Harper and Row, 1978. Chapter 10). These models may be useful for large populations, where the effects of chance can usually be ignored, but they simply are not realistic or useful for small populations. Most of the mathematical models that do allow for the variation caused by chance (called stochastic models) are either too simple to be realistic or too complex and restrictive to be of much use.

Monte Carlo Techniques and Basic's random function, RND, provided a solution to the problem. I first worked up a simple model that could be done by hand using the flipping of coins and a random numbers table. I used that for an ecology class laboratory exercise and gained enough experience with it to develop a larger, more complete computerized version. I developed the first computer program specifically for birds of prey (hawks, eagles, falcons and owls). It had two purposes: to stimulate and evaluate the probability of success of reintroducing peregrine falcons into the eastern United States, and to predict the fate of certain bald eagle populations, the species I have worked with mostly myself. The program had several options and permitted repeated simulations — something that a wildlife manager cannot do with the real thing. In real situations — wildlife transplants, reintroductions or the survival of a remnant population — there may be only one chance. In the computer the situation can be run hundreds or thousands of times to determine the likely range of outcomes.

After completing that program I decided to modify it to be useful for a wider range of species. The initial program assumed monogamous breeding (one male, one female); but many species are polygamous, so I added an option for that. The termin-

James Grier, Zoology Dept., North Dakota State University, Fargo, ND 58102.

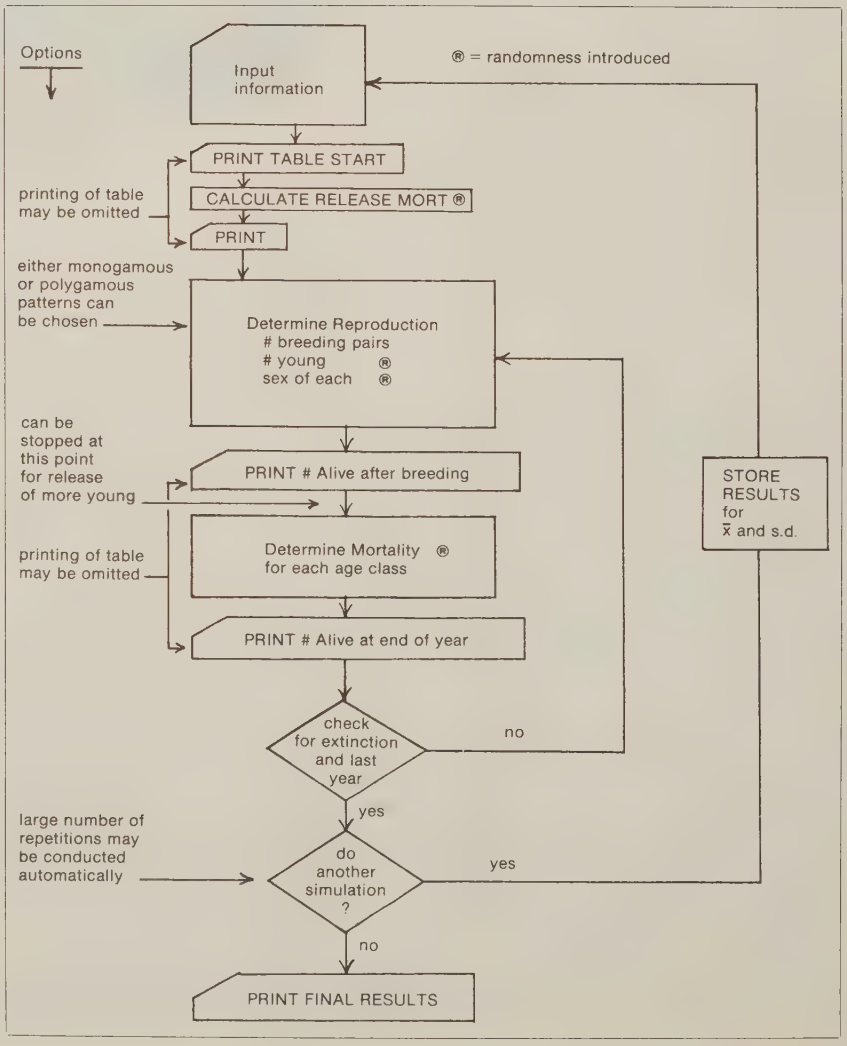
ology was changed to apply to all animals. The revised model still contains several more or less restrictive assumptions (Table 1), but it fits a large number of species and the program could be modified to make room for differences. The only real constraint in the model at present involves the number of young per female. The model gets increasingly awkward with increasing maximum numbers of young. Most species for which this model is intended produce only one to ten young at a time, and the program works fine. For species with larger numbers, the sections on reproduction (statements 620 to 730 and 2370 to 2640) could be modified. The flow chart is shown in Figure 1.

The revised program has now been used successfully in real applications, has become very popular for instructive purposes, and may even be used as a game. Life is most stimulating when one can mix work with pleasure, and this program has turned out to satisfy this ideal. By asking for information which often is not known, it points out and emphasizes the need for further research. In the meantime, and to better understand the model and what might happen under various situations, one can easily provide hypothetical values.

The program and some sample runs are provided here. The program contains about 300 statements and 10,000 characters. If that is too much for a user's available memory, it can be greatly shortened by separating out the comments and by eliminating some of the options, such as those that permit automatic repetition. I also have written 1) a companion deterministic program that shows what happens, in a similar format, when chance is removed and 2) a program (oriented toward birds) for life tables. Both of those programs are free from me on request. □

1) Breeding is seasonal and more or less synchronized in the population so that all birth or hatching occurs within a relatively short period.
2) If monogamous breeding is chosen, the number of breeding pairs is limited by the sex with the least number of adults; otherwise it is limited by the number of adult females, assuming that at least one male is present.
3) Members of potential pairs can find each other for purposes of breeding and are behaviorally/physiologically capable of breeding.
4) There are no inbreeding problems.
5) No density-dependent effects limit the population growth (although computer time might).
6) All of the mortality occurs outside the reproductive season. (And only those young that fledge or are weaned are counted in the first place. This eliminates those that die in the uterus, nest, etc. and permits the assumption to be fairly reasonable.)
7) Male and female mortality rates are the same.
8) Mortality rates are constant (within the random limits) for animals over one year of age. This assumes that the animals do not reach senility, and die from physiological old age in the wild.
9) The mortality and reproductive rates do not change significantly over time, that is, during the years of simulation.

Table 1. Assumptions of the Population Model.


Options
Input information
® = randomness introduced
PRINT TABLE START
printing of table may be omitted
CALCULATE RELEASE MORT ®
PRINT
either monogamous or polygamous patterns can be chosen
Determine Reproduction
breeding pairs
young ®
sex of each ®
can be stopped at this point for release of more young
PRINT # Alive after breeding
printing of table may be omitted
Determine Mortality ® for each age class
PRINT # Alive at end of year
check for extinction and last year
no
yes
large number of repetitions may be conducted automatically
do another simulation ?
yes
no
PRINT FINAL RESULTS
STORE RESULTS for x̄ and s.d.

```
END OF THIS SIMULATION. YOUR FINAL POPULATION
IS 2    PEREGRINE FALCON(S).

TABLE 3    PEREGRINE FALCONS ALIVE FOR EACH YEAR OF SIMULATION.

        FIRST     FIRST     SUBAD.    SUBAD.    ADULT     ADULT     TOTAL
        YEAR      YEAR
YEAR    MALES     FEMALES   MALES     FEMALES   MALES     FEMALES   ALIVE
-----------------------------------------------------------------------
   0        6         6         0         0         0         0        12
            0         1         0         0         0         0         1

   1        0         0         0         1         0         0         1
            0         0         0         1         0         0         1

   2        0         0         0         1         0         0         1
            0         0         0         0         0         0         0

-----------------------------------------------------------------------
THIS POPULATION OF PEREGRINE FALCONS DID NOT SURVIVE!

YOU RAN  3    SIMULATIONS.
THE MEAN FINAL POPULATION OF PEREGRINE FALCONS
(FOR SUCCESSFUL CASES) IS 1.5
WITH A STANDARD DEVIATION OF  .7    .
THERE WERE(WAS) 1    SIMULATION(S) THAT WENT TO EXTINCTION.

SEE YA.
TIME 4.6 SECS


RUN
7 SPG 10/22/79 13:54:31

  *** A STOCHASTIC MODEL FOR SMALL POPULATIONS ***

               BY JAMES W. GRIER
        ZOOLOGY DEPT.,  NDSU, FARGO ND 58105


DO YOU WANT INTRODUCTORY COMMENTS (1=YES, 2=NO)?
2
DO YOU WANT THE SIMULATION AUTOMATICALLY
REPEATED MORE THAN ONCE (1=YES, 2=NO)?
1
HOW MANY TIMES DO YOU WANT THE SIMULATION REPEATED?
100
(NOTE: IF THE PROGRAM STOPS ANYTIME BECAUSE YOU EXCEEDED
THE CPU LIMIT, HIT RETURN AND IT WILL RESUME.)
CHOOSE NOW WHETHER OR NOT TO HAVE THE TABLE PRINTED (1=YES,
PRINT THE TABLE; 2=NO, DO NOT PRINT THE TABLE AT ALL).?
2


WHAT SPECIES ARE YOU WORKING WITH?
PEREGRINE FALCON
DOES THIS SPECIES NORMALLY HAVE MONOGAMOUS (=M) OR
POLYGAMOUS (=P) BREEDING PATTERNS?
M
AT WHAT AGE DO YOU BELIEVE THE PEREGRINE FALCONS
NORMALLY BEGIN BREEDING?
3
WHAT IS THE MAXIMUM NUMBER OF YOUNG PER BREEDING FEMALE?
4
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 0    YOUNG?
20
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 1    YOUNG?
10
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 2    YOUNG?
30
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 3    YOUNG?
30
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 4    YOUNG?
10
WHAT IS THE ESTIMATED FIRST YEAR MORTALITY RATE
FOR PEREGRINE FALCONS?
60
WHAT IS YOUR GUESS FOR THE ANNUAL MORTALITY
RATE OF OLDER PEREGRINE FALCONS?
25
HOW MANY YEARS DO YOU WANT THE SIMULATION TO RUN?
5
HOW MANY MALE AND FEMALE OF THE YEAR ARE YOU
RELEASING (SPECIFY BOTH, SEPARATE WITH COMMA.)?
6,6
DO YOU PLAN TO RELEASE ANY OLDER PEREGRINE FALCONS (1=YES, 2=NO)?
2
WOULD YOU LIKE THE OPTION OF RELEASING ADDITIONAL YOUNG IN
SUBSEQUENT YEARS, AFTER THE SIMULATION HAS STARTED (1=YES,2=NO)
NOTE: DO NOT CHOOSE THIS OPTION IF THE SIMULATION IS BEING
      AUTOMATICALLY REPEATED MORE THAN 5 TIMES.
?
2

YOU RAN  100  SIMULATIONS.
THE MEAN FINAL POPULATION OF PEREGRINE FALCONS
(FOR SUCCESSFUL CASES) IS 2.639999
WITH A STANDARD DEVIATION OF  1.75    .
THERE WERE(WAS) 22   SIMULATION(S) THAT WENT TO EXTINCTION.
THIS GIVES AN ESTIMATED PROBABILITY OF EXTINCTION OF .22
OR THE CHANCE OF SUCCESS BEING .78  .

SEE YA.
TIME 7.0 SECS


 10 PRINT
 20 PRINT "  *** A STOCHASTIC MODEL FOR SMALL POPULATIONS ***"
 30 PRINT
 40 PRINT "               BY JAMES W. GRIER"
 50 PRINT "        ZOOLOGY DEPT., NDSU, FARGO ND 58105"
 60 PRINT
 70 PRINT
 80 PRINT "DO YOU WANT INTRODUCTORY COMMENTS (1=YES, 2=NO)";
 90 INPUT A6
100 IF A6=2 THEN 290
110 PRINT "     CHANCE EVENTS ARE IMPORTANT FACTORS IN THE GROWTH PATTERNS"
120 PRINT "OF ANIMAL POPULATIONS, PARTICULARLY SMALL POPULATIONS.  THIS"
130 PRINT "PROGRAM INCORPORATES CHANCE FOR BROOD OR LITTER SIZE, SEX OF"
140 PRINT "EACH INDIVIDUAL BORN OR HATCHED, AND MORTALITY -- AT PROBABILITIES"
150 PRINT "FOR REPRODUCTION AND MORTALITY THAT YOU SUPPLY. (PROBABILITY"
160 PRINT "FOR SEX RATIO IS ASSUMED TO BE 50:50.)"
170 PRINT
180 PRINT "      THE PROGRAM IS BOTH GENERAL AND FLEXIBLE, PERMITTING"
190 PRINT "SEVERAL OPTIONS OR CHOICES. ONE OPTION ALLOWS THE SIMULATION TO BE"
200 PRINT "REPEATED MORE THAN ONCE AUTOMATICALLY.  DEPENDING ON THE SIZE OF"
210 PRINT "THE POPULATION, LENGTH OF SIMULATION, AND OTHER CHARACTERISTICS"
220 PRINT "OF THE SET OF PARAMETERS THAT YOU USE, HOWEVER, THE SIMULATION"
230 PRINT "MAY REQUIRE A LOT OF COMPUTER TIME AND A LARGE NUMBER OF"
240 PRINT "REPETITIONS COULD USE AN EXCESSIVE AMOUNT OF COMPUTER TIME."
250 PRINT "THEREFORE, YOU SHOULD NOT CHOOSE TO USE THIS OPTION UNTIL AFTER"
260 PRINT "YOU HAVE RUN A SIMILAR SIMULATION AT LEAST ONCE, CHECKED THE"
270 PRINT "TIME IT TOOK AND ESTIMATED THE TIME REQUIREMENTS FOR SEVERAL"
280 PRINT "REPETITIONS."
290 PRINT "DO YOU WANT THE SIMULATION AUTOMATICALLY"
300 PRINT "REPEATED MORE THAN ONCE (1=YES, 2=NO)";
310 C=0
320 INPUT A:
330 IF A0=2 THEN 360
340 PRINT "HOW MANY TIMES DO YOU WANT THE SIMULATION REPEATED";
350 INPUT C
355 PRINT "(NOTE: IF THE PROGRAM STOPS ANYTIME BECAUSE YOU EXCEEDED"
356 PRINT "THE CPU LIMIT, HIT RETURN AND IT WILL RESUME.)"
360 IF A6=2 THEN 480
370 PRINT
371 PRINT "      THE INPUT IS REQUESTED FROM THE CONTEXT OF A POPULATION"
372 PRINT "BEING ARTIFICIALLY INTRODUCED TO AN AREA, AS IN THE INTRODUCTION"
373 PRINT "OF AN EXOTIC SPECIES OR THE REINTRODUCTION OR TRANSPLANTING OF"
374 PRINT "AN ENDANGERED SPECIES.  THE PROGRAM CAN ALSO BE USED, HOWEVER,"
375 PRINT "TO MODEL EXISTING NATURAL POPULATIONS.  WHEN THE PROGRAM REQUESTS"
376 PRINT 'INFORMATION ON ANIMALS BEING "RELEASED," SIMPLY THINK IN TERMS'
377 PRINT 'OF ANIMALS THAT "ARE PRESENT."'
378 PRINT
380 PRINT "      NEXT, AN EXPLANATION OF THE RESULTS:  THE OUTCOME IS"
390 PRINT "TABULATED ON A YEAR BY YEAR BASIS. THERE ARE TWO LINES PRINTED"
400 PRINT "FOR EACH YEAR. THE FIRST IS THE NUMBER OF ANIMALS ALIVE AT THE"
410 PRINT "END OF THE REPRODUCTIVE SEASON.  THE SECOND IS THE NUMBER ALIVE"
420 PRINT "AT THE END OF THE YEAR, AFTER MORTALITY HAS OCCURRED AND PRIOR"
430 PRINT "TO THE NEXT REPRODUCTIVE SEASON. THE FINAL OUTCOME IS STORED"
440 PRINT "AND AVERAGED OVER SUCCESSIVE SIMULATIONS.  IF THERE ARE MORE"
450 PRINT "THAN ONE.  IN CASE THE SIMULATION IS REPEATED MORE THAN FIVE "
460 PRINT "TIMES, THE YEARLY TABLE IS PRINTED ONLY THE FIRST FIVE TIMES,"
470 PRINT "UNLESS YOU DO NOT WANT THE TABLE PRINTED AT ALL.  YOU NEED TO"
480 PRINT "CHOOSE NOW WHETHER OR NOT TO HAVE THE TABLE PRINTED (1=YES,"
490 PRINT "PRINT THE TABLE; 2=NO, DO NOT PRINT THE TABLE AT ALL).";
500 INPUT A2
510 IF A6=2 THEN 580
520 PRINT
530 PRINT "      A FINAL NOTE BEFORE STARTING:  THE MODEL NECESSARILY"
540 PRINT "CONTAINS SEVERAL MATHEMATICAL AND BIOLOGICAL ASSUMPTIONS BUT"
550 PRINT "IT WOULD TAKE TOO MUCH SPACE TO LIST THEM ALL HERE. IF YOU"
560 PRINT "WANT A LIST OF THESE OR ANY ADDITIONAL DOCUMENTATION, OR TO"
570 PRINT "REPORT ANY BUGS, PLEASE CONTACT J. W. GRIER."
580 PRINT
590 PRINT
600 PRINT "WHAT SPECIES ARE YOU WORKING WITH";
610 INPUT S$
620 PRINT "DOES THIS SPECIES NORMALLY HAVE MONOGAMOUS (=M) OR"
630 PRINT "POLYGAMOUS (=P) BREEDING PATTERNS";
640 INPUT B$
650 PRINT "AT WHAT AGE DO YOU BELIEVE THE "; S$; "S"
660 PRINT "NORMALLY BEGIN BREEDING";
```

```
670 INPUT A
680 PRINT "WHAT IS THE MAXIMUM NUMBER OF YOUNG PER BREEDING FEMALE";
690 INPUT B
700 FOR L=0 TO B
710 PRINT "WHAT PERCENT OF THE ADULT FEMALES PRODUCE";L;"YOUNG";
720 INPUT B(L+1)
730 NEXT L
740 PRINT "WHAT IS THE ESTIMATED FIRST YEAR MORTALITY RATE"
750 PRINT "FOR "; S$;"S";
760 INPUT I
770 PRINT "WHAT IS YOUR GUESS FOR THE ANNUAL MORTALITY"
780 PRINT "RATE OF OLDER ";S$;"S";
790 INPUT H
800 LET S=0
810 LET S2=0
820 LET G=0
830 LET R=0
840 PRINT "HOW MANY YEARS DO YOU WANT THE SIMULATION TO RUN";
850 INPUT N
860 DIM M(35), F(35)
870 DIM E(35), J(35)
880 MAT M=ZER(35)
890 MAT F=ZER(35)
900 MAT E=ZER(35)
910 MAT J=ZER(35)
920 IF A=1 THEN 1010
930 PRINT "HOW MANY MALE AND FEMALE OF THE YEAR ARE YOU"
940 PRINT "RELEASING (SPECIFY BOTH. SEPARATE WITH COMMA.)";
950 INPUT E(A),J(A)
960 LET U(1)=0
970 LET U(2)=0
980 PRINT "DO YOU PLAN TO RELEASE ANY OLDER ";S$;"S (1=YES, 2=NO)";
990 INPUT A3
1000 IF A3=2 THEN 1100
1010 FOR X = 1 TO A-2
1020 PRINT "HOW MANY MALES AND FEMALES ";X;"YEAR(S) OLD";
1030 INPUT E(A-X),J(A-X)
1040 LET U(1)=U(1)+E(A-X)
1050 LET U(2)=U(2)+J(A-X)
1060 NEXT X
1070 PRINT "HOW MANY MALES AND FEMALES";" ";A-1;" YEARS"
1080 PRINT "OR OLDER (POTENTIAL BREEDERS FOR THE NEXT SEASON) ARE YOU RELEASING";
1090 INPUT E(1),J(1)
1100 PRINT "WOULD YOU LIKE THE OPTION OF RELEASING ADDITIONAL YOUNG IN"
1110 PRINT "SUBSEQUENT YEARS, AFTER THE SIMULATION HAS STARTED (1=YES,2=NO)"
1120 PRINT "NOTE: DO NOT CHOOSE THIS OPTION IF THE SIMULATION IS BEING"
1130 PRINT "      AUTOMATICALLY REPEATED MORE THAN 5 TIMES."
1140 INPUT A1
1150 IF A1=2 THEN 1200
1160 PRINT "AFTER THE REPRODUCTION FOR THE ANIMALS IN THE WILD HAS BEEN"
1170 PRINT "DETERMINED, THE SIMULATION WILL STOP, AFTER THE QUESTION MARK"
1180 PRINT "APPEARS, ENTER THE NUMBER OF MALE, FEMALE YOUNG OF THE YEAR THAT"
1190 PRINT "YOU WISH TO ADD FOR THAT YEAR."
1200 MAT M=E
1210 MAT F=J
1220 LET D(1)=0
1230 LET D(2)=0
1240 MAT D=U
1250 LET D(1)=D(1)+M(1)
1260 LET D(2)=D(2)+F(1)
1270 IF R>4 THEN 1320
1280 IF A2=2 THEN 1320
1290 PRINT
1300 PRINT "TABLE";R+1;S$;"S ALIVE FOR EACH YEAR OF SIMULATION."
1310 PRINT
1320 LET W=0
1330 LET O(1)=0
1340 LET O(2)=0
1350 LET Y=A
1360 IF R>4 THEN 1430
1370 IF A2=2 THEN 1430
1380 '#####   ########  ########  ########  ########  ########  ########
1390 PRINT USING 1380," ","FIRST","FIRST","SUBAD.","SUBAD.","ADULT","ADULT","TOTAL"
1400 PRINT USING "1380," "," ","YEAR","YEAR"," "," "," "," "
1410 PRINT USING 1380,"YEAR","MALES","FEMALES","MALES","FEMALES","MALES","FEMALES", "ALIVE"
1420 PRINT "-------------------------------------------------------------------------"
1430 IF W=0 THEN 2750
1440 LET P=M(Y)
1450 IF P=0 THEN 1500
1460 FOR Q=1 TO P
1470 IF INT(100*RND)>=I THEN 1490
1480 LET M(Y)=M(Y)-1
1490 NEXT Q
1500 LET P=F(Y)
1510 IF P=0 THEN 1560
1520 FOR Q=1 TO P
1530 IF INT(100*RND)>=I THEN 1550
1540 LET F(Y)=F(Y)-1
1550 NEXT Q
1560 FOR Z=Y-1 TO 1 STEP -1
1570 LET P=M(Z)
1580 IF P=0 THEN 1630
1590 FOR  Q=1 TO P
1600 IF INT(100*RND)>=H THEN 1620
1610 LET M(Z)=M(Z)-1
1620 NEXT Q
1630 LET P =F(Z)
1640 IF P=0 THEN 1690
1650 FOR Q=1 TO P
1660 IF INT(100*RND)>=H THEN 1680
1670 LET F(Z)=F(Z)-1
1680 NEXT Q
1690 NEXT Z
1700 GOSUB 1720
1710 GO TO 1900
1720 LET D(1)=0
1730 FOR L=Y-A+1 TO Y-1
1740 LET D(1)=D(1) + M(L)
1750 NEXT L
1760 LET D(2)=0
1770 FOR L=Y-A+1 TO Y-1
1780 LET D(2)=D(2)+F(L)
1790 NEXT L
1800 LET O(1)=0
1810 FOR L=1 TO Y-A
1820 LET O(1) =O(1) +M(L)
1830 NEXT L
1840 LET O(2)=0
1850 FOR L=1 TO Y-A
1860 LET O(2) = O(2) + F(L)
1870 NEXT L
1880 LET T=M(Y)+F(Y)+D(1)+D(2)+O(1)+O(2)
1890 RETURN
1900 IF R>4 THEN 1940
1910 IF A2=2 THEN 1940
1920 PRINT USING 1380," ",M(Y),F(Y),D(1),D(2),O(1),O(2),M(Y)+F(Y)+D(1)+D(2)+O(1)+O(2)
1930 PRINT
1940 IF T<>0 THEN 2010
1950 LET G=G+1
1960 IF R>4 THEN 2000
1970 IF A2=2 THEN 2000
1980 PRINT "-------------------------------------------------------------------------"
1990 PRINT "THIS POPULATION OF ";S$;"S DID NOT SURVIVE "
2000 GO TO 2070
2010 IF W<=N THEN 2350
2020 IF R>4 THEN 2070
2030 IF A2=2 THEN 2070
2040 PRINT "-------------------------------------------------------------------------"
2050 PRINT "END OF THIS SIMULATION. YOUR FINAL POPULATION"
2060 PRINT "IS";T;S$;"(S)."
2070 LET R=R+1
2080 LET S=S+T
2090 LET S2=S2+T*T
2100 IF C=0 THEN 2130
2110 IF R<C THEN 1200
2120 IF C>0 THEN 2160
2130 PRINT "WOULD YOU LIKE TO RUN ANOTHER SIMULATION (1=YES, 2=NO)";
2140 INPUT A4
2150 IF A4=1 THEN 2300
2160 GO TO 2200
2170 PRINT
2180 PRINT "SEE YA."
2190 STOP
2200 IF R=1 THEN 2170
2210 PRINT
2220 PRINT "YOU RAN ";R;"SIMULATIONS."
2222 IF (R-G)=0 THEN 2250
2223 IF ((R-G)-1)=0 THEN 2250
2225 PRINT "THE MEAN FINAL POPULATION OF ";S$;"S"
2230 PRINT "((FOR SUCCESSFUL CASES) IS";INT(100*S/(R-G))/100
2240 PRINT "WITH A STANDARD DEVIATION OF ";INT(100*SQR(((R-G)*S2-S*S)/(R-G)/((R-G)-1)))/
2250 PRINT "THERE WERE(WAS)";G;"SIMULATION(S) THAT WENT TO EXTINCTION."
2260 IF R<20 THEN 2290
2270 PRINT "THIS GIVES AN ESTIMATED PROBABILITY OF EXTINCTION OF";INT(100*G/R)/100
2280 PRINT "OR THE CHANCE OF SUCCESS BEING";1-(INT(100*G/R)/100);"."
2290 GO TO 2170
2300 PRINT "WOULD YOU LIKE TO HAVE THE NEXT SIMULATION RUN UNDER"
2310 PRINT "THE SAME CONDITIONS AS THE LAST TIME (1=YES, 2=NO)";
2320 INPUT A5
2330 IF A5=2 THEN 600
2340 GO TO 1200
2350 LET Y=Y+1
2360 GOSUB 1720
2370 LET K=0
2380 IF B$="M" THEN 2410
2390 IF O(1)=0 THEN 2650
2400 GO TO 2420
2410 IF O(1)<O(2) THEN 2550
2420 FOR J=1 TO O(2)
2430 GO SUB 2450
2440 GO TO 2520
2450 LET P=INT((B+1)*RND)
2460 FOR L=0 TO B
2470 IF P<>L THEN 2500
2480 IF INT(100*RND)>=(B(L+1)) THEN 2450
2490 LET K=K+P
2500 NEXT L
2510 RETURN
2520 NEXT J
2530 IF B$="P" THEN 2590
2540 GO TO 2590
2550 FOR J=1 TO O(1)
2560 GO SUB 2450
2570 GO TO 2580
2580 NEXT J
2590 FOR L=1 TO K
2600 IF INT(2*RND)=0 THEN 2630
2610 LET M(Y)=M(Y) + 1
2620 GO TO 2640
2630 LET F(Y)=F(Y)+1
2640 NEXT L
2650 IF R>4 THEN 2680
2660 IF A2=2 THEN 2680
2670 PRINT USING 1380,W,M(Y),F(Y),D(1),D(2),O(1),O(2),M(Y)+F(Y)+D(1)+D(2)+O(1)+O(2)
2680 LET W=W+1
2690 IF A1<>1 THEN 1440
2700 IF W=1 THEN 1440
2710 INPUT U,V
2720 LET M(Y)=M(Y)+U
2730 LET F(Y)=F(Y)+V
2740 GO TO 1440
2750 IF A=1 THEN 2770
2760 GO TO 2650
2770 IF R>4 THEN 2680
2780 IF A2=2 THEN 2680
2790 PRINT USING 1380, W,M(Y),F(Y),0,0,0,0,M(Y)+F(Y)
2800 GO TO 2680
2810 END
```

```
RUN
7 SPG 10/22/79 13:48:14

  *** A STOCHASTIC MODEL FOR SMALL POPULATIONS ***

              BY JAMES W. GRIER
        ZOOLOGY DEPT., NDSU, FARGO ND 58105


DO YOU WANT INTRODUCTORY COMMENTS (1=YES, 2=NO)?
1
     CHANCE EVENTS ARE IMPORTANT FACTORS IN THE GROWTH PATTERNS
OF ANIMAL POPULATIONS, PARTICULARLY SMALL POPULATIONS.  THIS
PROGRAM INCORPORATES CHANCE FOR BROOD OR LITTER SIZE, SEX OF
EACH INDIVIDUAL BORN OR HATCHED, AND MORTALITY -- AT PROBABILITIES
FOR REPRODUCTION AND MORTALITY THAT YOU SUPPLY. (PROBABILITY
FOR SEX RATIO IS ASSUMED TO BE 50:50.)

     THE PROGRAM IS BOTH GENERAL AND FLEXIBLE, PERMITTING
SEVERAL OPTIONS OR CHOICES. ONE OPTION ALLOWS THE SIMULATION TO BE
REPEATED MORE THAN ONCE AUTOMATICALLY.  DEPENDING ON THE SIZE OF
THE POPULATION, LENGTH OF SIMULATION, AND OTHER CHARACTERISTICS
OF THE SET OF PARAMETERS THAT YOU USE, HOWEVER, THE SIMULATION
MAY REQUIRE A LOT OF COMPUTER TIME AND A LARGE NUMBER OF
REPETITIONS COULD USE AN EXCESSIVE AMOUNT OF COMPUTER TIME.
THEREFORE, YOU SHOULD NOT CHOOSE TO USE THIS OPTION UNTIL AFTER
YOU HAVE RUN A SIMILAR SIMULATION AT LEAST ONCE, CHECKED THE
TIME IT TOOK AND ESTIMATED THE TIME REQUIREMENTS FOR SEVERAL
REPETITIONS.
DO YOU WANT THE SIMULATION AUTOMATICALLY
REPEATED MORE THAN ONCE (1=YES, 2=NO)?
1
HOW MANY TIMES DO YOU WANT THE SIMULATION REPEATED?
3
(NOTE: IF THE PROGRAM STOPS ANYTIME BECAUSE YOU EXCEEDED
THE CPU LIMIT, HIT RETURN AND IT WILL RESUME.)

     THE INPUT IS REQUESTED FROM THE CONTEXT OF A POPULATION
BEING ARTIFICIALLY INTRODUCED TO AN AREA, AS IN THE INTRODUCTION
OF AN EXOTIC SPECIES OR THE REINTRODUCTION OR TRANSPLANTING OF
AN ENDANGERED SPECIES.  THE PROGRAM CAN ALSO BE USED, HOWEVER,
TO MODEL EXISTING NATURAL POPULATIONS.  WHEN THE PROGRAM REQUESTS
INFORMATION ON ANIMALS BEING 'RELEASED,' SIMPLY THINK IN TERMS
OF ANIMALS THAT 'ARE PRESENT.'

     NEXT, AN EXPLANATION OF THE RESULTS:  THE OUTCOME IS
TABULATED ON A YEAR BY YEAR BASIS. THERE ARE TWO LINES PRINTED
FOR EACH YEAR, THE FIRST IS THE NUMBER OF ANIMALS ALIVE AT THE
END OF THE REPRODUCTIVE SEASON. THE SECOND IS THE NUMBER ALIVE
AT THE END OF THE YEAR, AFTER MORTALITY HAS OCCURRED AND PRIOR
TO THE NEXT REPRODUCTIVE SEASON. THE FINAL OUTCOME IS STORED
AND AVERAGED OVER SUCCESSIVE SIMULATIONS, IF THERE ARE MORE
THAN ONE. IN CASE THE SIMULATION IS REPEATED MORE THAN FIVE
TIMES, THE YEARLY TABLE IS PRINTED ONLY THE FIRST FIVE TIMES,
UNLESS YOU DO NOT WANT THE TABLE PRINTED AT ALL.  YOU NEED TO
CHOOSE NOW WHETHER OR NOT TO HAVE THE TABLE PRINTED (1=YES,
PRINT THE TABLE; 2=NO, DO NOT PRINT THE TABLE AT ALL).?
1

     A FINAL NOTE BEFORE STARTING:  THE MODEL NECESSARILY
CONTAINS SEVERAL MATHEMATICAL AND BIOLOGICAL ASSUMPTIONS BUT
IT WOULD TAKE TOO MUCH SPACE TO LIST THEM ALL HERE. IF YOU
WANT A LIST OF THESE OR ANY ADDITIONAL DOCUMENTATION, OR TO
REPORT ANY BUGS, PLEASE CONTACT J. W. GRIER.


WHAT SPECIES ARE YOU WORKING WITH?
PEREGRINE FALCON
DOES THIS SPECIES NORMALLY HAVE MONOGAMOUS (=M) OR
POLYGAMOUS (=P) BREEDING PATTERNS?
M
AT WHAT AGE DO YOU BELIEVE THE PEREGRINE FALCONS
NORMALLY BEGIN BREEDING?
3
WHAT IS THE MAXIMUM NUMBER OF YOUNG PER BREEDING FEMALE?
4
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 0    YOUNG?
20
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 1    YOUNG?
10
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 2    YOUNG?
30
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 3    YOUNG?
30
WHAT PERCENT OF THE ADULT FEMALES PRODUCE 4    YOUNG?
10
WHAT IS THE ESTIMATED FIRST YEAR MORTALITY RATE
FOR PEREGRINE FALCONS?
60
WHAT IS YOUR GUESS FOR THE ANNUAL MORTALITY
RATE OF OLDER PEREGRINE FALCONS?
25
HOW MANY YEARS DO YOU WANT THE SIMULATION TO RUN?
5
HOW MANY MALE AND FEMALE OF THE YEAR ARE YOU
RELEASING (SPECIFY BOTH, SEPARATE WITH COMMA.)?
6,6
DO YOU PLAN TO RELEASE ANY OLDER PEREGRINE FALCONS (1=YES, 2=NO)?
2
WOULD YOU LIKE THE OPTION OF RELEASING ADDITIONAL YOUNG IN
SUBSEQUENT YEARS, AFTER THE SIMULATION HAS STARTED (1=YES,2=NO)
NOTE: DO NOT CHOOSE THIS OPTION IF THE SIMULATION IS BEING
      AUTOMATICALLY REPEATED MORE THAN 5 TIMES.
?
2

TABLE 1    PEREGRINE FALCONS ALIVE FOR EACH YEAR OF SIMULATION.

                FIRST          FIRST          SUBAD.         SUBAD.         ADULT          ADULT          TOTAL
YEAR            YEAR MALES     YEAR FEMALES   MALES          FEMALES        MALES          FEMALES        ALIVE
-------------------------------------------------------------------------------------------------------------------
   0                 6              6              0              0              0              0             12
                     1              3              0              0              0              0              4

   1                 0              0              1              3              0              0              4
                     0              0              0              3              0              0              3

   2                 0              0              0              3              0              0              3
                     0              0              0              3              0              0              3

   3                 0              0              0              0              0              3              3
                     0              0              0              0              0              1              1

   4                 0              0              0              0              0              1              1
                     0              0              0              0              0              1              1

   5                 0              0              0              0              0              1              1
                     0              0              0              0              0              1              1
-------------------------------------------------------------------------------------------------------------------
END OF THIS SIMULATION. YOUR FINAL POPULATION
IS 1    PEREGRINE FALCON(S).

TABLE 2    PEREGRINE FALCONS ALIVE FOR EACH YEAR OF SIMULATION.

                FIRST          FIRST          SUBAD.         SUBAD.         ADULT          ADULT          TOTAL
YEAR            YEAR MALES     YEAR FEMALES   MALES          FEMALES        MALES          FEMALES        ALIVE
-------------------------------------------------------------------------------------------------------------------
   0                 6              6              0              0              0              0             12
                     4              4              0              0              0              0              8

   1                 0              0              4              4              0              0              8
                     0              0              2              3              0              0              5

   2                 0              0              2              3              0              0              5
                     0              0              1              2              0              0              3

   3                 1              0              0              0              1              2              4
                     0              0              0              0              1              2              3

   4                 1              1              0              0              1              2              5
                     0              0              0              0              0              2              2

   5                 0              0              0              0              0              2              2
                     0              0              0              0              0              2              2
-------------------------------------------------------------------------------------------------------------------
```

NICHE: A Basic Game of Ecology

James D. Lehman

During the 1977 fall semester at Purdue University, I participated in Dr. Franz J. Frederick's Computer Assisted Instruction course. While initially quite apprehensive about this, my first encounter with the computer, it turned out to be not only informative but highly enjoyable as well. I was soon captivated by the potentials of CAI.

Background

Before long it became apparent to me that, while it might not be an educational panacea, the computer, especially the small computer, would definitely become an integral part of the classroom of the not-too-distant future. So, I began to consider ways in which the computer could be beneficially applied to the course for which I serve as graduate instructor at Purdue, Biology for Elementary School Teachers.

At the time, my course was involved in the important biological notions of ecosystem and population. Upon invesigation, I found that some relevant computer applications, notably population dynamics and genetics simulations, had been done. For the most part, however, these tended to be too detailed or boring (or both) for my introductory course. Therefore, I decided to try to develop a relatively elementary game/simulation of ecological orientation.

I chose the game/simulation format because it offered a means of presenting some of the complexities of ecology in an enjoyable fashion. While the use of computer game-playing as an educational tool has been criticized, I think it is difficult to dispute its intrinsic motivational value. After all, games are fun! This, coupled with the prospect of to some extent simulating ecological processes, made the idea an attractive one.

James D. Lehman, Dept. of Biological Sciences, Purdue University, West Lafayette, IN 47907.

```
10 REM**** NICHE--AN ECOLOGICAL GAME
20 REM**** JAMES D. LEHMAN, DEPT. OF BIOLOGICAL SCIENCES
30 REM**** PURDUE UNIVERSITY, W. LAFAYETTE, INDIANA 47906
40 RANDOMIZE
50 T=0\K=0\P=100.
60 &\&\& ">>>>>>>>>>  NICHE  <<<<<<<<<<"\&
70 & TAB(5);"AN ECOLOGICAL GAME"\&\&\&
80 & "WELCOME TO THE GAME OF NICHE!"
90 & "DO YOU NEED INSTRUCTIONS (Y/N) ";
100 INPUT I$
110 ON ERROR GOTO 90
120 IF I$="N" OR I$="NO" THEN 520
130 &\&\& "INSTRUCTIONS FOR NICHE---"\&
140 & "NICHE REFERS TO ALL OF THE ECOLOGICAL VARIABLES WHICH"
150 & "RELATE TO A GIVEN ORGANISM--ITS HABITAT, LIVING SPACE,"
160 & "AND ROLE IN THE ECOSYSTEM.  IN THE GAME OF NICHE YOUR JOB"
170 & "IS TO FIT A SELECTED ORGANISM INTO ITS NICHE.  YOU SELECT"
180 & "ONE OF 5 ORGANISMS AND RECEIVE A SMALL POPULATION (100"
190 & "INDIVIDUALS) OF THE ORGANISMS.  YOUR GOAL IN NICHE IS TO"
200 & "MAXIMIZE THE SIZE OF YOUR POPULATION BY PROVIDING IDEAL"
210 & "CONDITIONS FOR GROWTH.  AT THE BEGINNING OF THE GAME (TURN 1)"
220 & "AND AT EACH SUBSEQUENT TURN YOU WILL INPUT THE"
230 & "ENVIRONMENT, RANGE, AND COMPETITOR FOR YOUR POPULATION."
240 & "ONCE DURING THE GAME, YOU MAY INPUT THE PREDATOR KILL OPTION."
250 & "THIS WILL REDUCE BUT NOT ELIMINATE THE EFFECTS OF PREDATION"
260 & "ON YOUR POPULATION FOR THAT TURN ONLY.  AT THE END OF EACH TURN,"
270 & "THE FOLLOWING INFORMATION WILL BE OUTPUT:  THE AMOUNT OF"
280 & "FOOD PRESENT IN THE RANGE, THE AMOUNT OF FOOD TAKEN BY"
290 & "COMPETITORS, THE AMOUNT OF FOOD AVAILABLE TO YOUR POPULATION,"
300 & "DEATHS DUE TO PREDATION, DEATHS DUE TO STARVATION AND OTHER"
310 & "CAUSES, BIRTHS, AND THE NEW TOTAL POPULATION.  OF COURSE,"
320 & "WHAT YOU INPUT WILL AFFECT THE RESULTS WHICH ARE OUTPUT."
330 &\& "TO CONTINUE INSTRUCTIONS, TYPE GO"
340 INPUT G$
350 ON ERROR GOTO 330
360 IF G$<>"GO" THEN 90
370 &\&\& "FOR INSTANCE, IF THE INPUT RANGE IS TOO SMALL FOR YOUR"
380 & "POPULATION, THERE WILL BE INSUFFICIENT FOOD (100 UNITS PER"
390 & "ORGANISM IS OPTIMUM) AND SOME STARVATION WILL RESULT."
400 & "IF THE INPUT RANGE IS TOO LARGE, THERE WILL BE MANY PREDATORS"
410 & "AND DEATHS DUE TO PREDATION WILL BE HIGH.  THE GAME LASTS"
420 & "FOR 5 TURNS, EACH OF WHICH CORRESPONDS TO ABOUT 1 TO 2"
430 & "GENERATIONS OF YOUR ORGANISM.  YOU SHOULD RESPOND TO INPUT"
440 & "PROMPTS BY TYPING WHAT IS IN THE PARENTHESES FOLLOWING YOUR"
450 & "CHOICE.  YOU MAY INITIATE EACH NEW TURN BY TYPING GO."
460 & "GOOD LUCK!! AND, MAY THE BEST ECOLOGIST WIN!!"
480 &\&\& "TO BEGIN THE GAME, TYPE GO"
490 INPUT G$
500 ON ERROR GOTO 480
510 IF G$<>"GO" THEN 90
520 &\&\& "HERE WE GO!"\&\&
530 REM**** SELECTION OF THE ORGANISM
540 & "WHAT ORGANISM---"
550 & TAB(5);"PARAMECIUM CAUDATUM (1)"
560 & TAB(5);"RAINBOW TROUT (2)"
570 & TAB(5);"ANTEATER (3)"
580 & TAB(5);"GRAY SQUIRREL (4)"
590 & TAB(5);"KANGAROO RAT (5)"
600 INPUT O
610 ON ERROR GOTO 640
630 GOTO 650
```

The Concept

I decided to center the game around the biological concept of a niche. In ecology, a niche is the place in the scheme of things occupied by an organism. It really refers to all of the ecological variables which relate to a given organism. Thus, the niche of any one organism is dependent upon a multiplicity of interacting factors. So, the idea of a niche is one which is well suited to the capabilities of a computer.

The Game

The object of the game NICHE is to fit one of five given organisms into its proper niche. The player attempts this by specifying the environment, range, and competitor for a small population of these organisms. Assuming no catastrophes, the game lasts for five generations of the organism. If conditions are well specified, the population will bountifully multiply in an ideal niche of plentiful food, low predation, and little competition. A poorly specified niche could have disastrous consequences for the population. Of course, as in a real ecosystem, the factors in NICHE are interrelated.

From a biological point of view, NICHE is not intended to be an accurate representation of an ecosystem. Rather, it is merely intended to be a fun introduction to the kinds of interactions found in real ecosystems. NICHE requires no special knowledge of biology; anyone can play!

On the computing end, NICHE was designed in BASIC-PLUS to run on Purdue's PDP-11/70. However, it should run with little or no modification on most BASIC systems. Since NICHE was essentially a beginning effort, there are probably areas in the program which could be tightened up. However, I think the basic program structure is fairly sound. NICHE could be a useful part of an introductory biology program or just an interesting addition to the hobbyist's collection. Enjoy it!

```
640 % "IMPROPER RESPONSE--TRY AGAIN."\GOTO 540
650 REM**** SET UP TURN COUNTER AND PRINT TURN
660 T=T+1\%\%\% "TURN "#T\%\%
670 REM**** SELECTION OF ENVIRONMENT
680 % "WHAT ENVIRONMENT---"
690 % TAB(5)#"FRESH WATER POND (P)"
700 % TAB(5)#"FRESH WATER LAKE (L)"
710 % TAB(5)#"TROPICAL RAIN FOREST (R)"
720 % TAB(5)#"BOREAL CONIFEROUS FOREST (C)"
730 % TAB(5)#"TEMPERATE DECIDUOUS FOREST (T)"
740 % TAB(5)#"NORTH AMERICAN DESERT (D)"
750 % TAB(5)#"TEMPERATE GRASSLAND (G)"
760 INPUT E$
770 ON ERROR GOTO 800
790 GOTO 810
800 % "IMPROPER RESPONSE--TRY AGAIN."\GOTO 680
810 REM**** GOSUB TO SET VALUES OF E, E1, AND U$
820 GOSUB 3000
830 REM**** SELECTION OF RANGE SIZE
840 %\%\% "WHAT RANGE FOR THE WHOLE POPULATION IN "#U$#" ---"
844 IF U$="HECTARES" THEN % "(1 HECTARE = 10000 SQUARE METERS)"
846 IF U$="DEKASTERES" THEN % "(1 DEKASTERE = 10 CUBIC METERS)"
850 INPUT R
860 ON ERROR GOTO 890
870 IF R<=0 THEN 890
880 GOTO 900
890 % "IMPROPER RESPONSE--TRY AGAIN."\GOTO 840
900 REM**** SELECTION OF COMPETITOR (ORGANISM DEPENDENT)
910 REM**** AND SETTING OF TRUE OPTIMUM RANGE
920 %\%\% "WHAT COMPETITOR---"
930 IF O=1 THEN 4000
940 IF O=2 THEN 4100
950 IF O=3 THEN 4200
960 IF O=4 THEN 4300
970 IF O=5 THEN 4400
980 REM**** GOSUB TO SET VALUES OF R1 AND R2
990 GOSUB 5000
1000 REM**** SET C1 VALUE
1010 IF C$="C" OR C$="E" OR C$="G" OR C$="I" OR C$="K" OR C$="M" THEN C1=0
1020 IF C$="B" OR C$="D" OR C$="L" OR C$="N" THEN C1=.2
1030 IF C$="F" OR C$="J" THEN C1=.4
1040 IF C$="A" OR C$="H" OR C$="O" THEN C1=.6
1050 REM**** CHECK FOR PREVIOUS USE OF PREDATOR KILL
1060 IF K=1 THEN 1140
1070 %\%\% "PREDATOR KILL (Y/N)---"
1080 INPUT K$
1090 IF K$="Y" OR K$="YES" THEN 1120
1100 IF K$="N" OR K$="NO" THEN 1140
1110 % "IMPROPER RESPONSE--TRY AGAIN."\GOTO 1070
1120 REM**** SET COUNTER K AND SET D1 AT A LOW LEVEL
1130 K=1 \ D1=INT(.02*P) \ GOTO 1160
1140 REM**** SET D1 AT THE NORMAL LEVEL
1150 D1=INT((.25*RND(1)+.1)*P) + INT(R2*P)
1160 REM**** SET VARIABLE VALUES
1170 R1=R/R3 \ F1=INT(100*R1*P) \ F2=INT(C1*F1)
1180 F3=F1-F2 \ F4=F3/(100*P)
1184 IF F4>1 THEN F4=1
1186 D2=INT((1-F4+E1)*P)
1190 REM**** GOSUB TO SET F5 VALUE
1200 GOSUB 6000
1210 REM**** SET BIRTHS
1220 B=INT(F5*P)
1230 IF E1=1 THEN 1250
1240 GOTO 1260
1250 B=0 \ D1=0 \ D2=P
1260 REM**** SET POPULATION
1270 P1=P \ P=P+B-D1-D2
1280 IF P<0 THEN 1300
1290 GOTO 1340
1300 X=RND(1) \ Y=RND(1)
1310 IF P1+B=INT(D1*X)+INT(D2*Y) THEN 1330
1320 GOTO 1300
1330 D1=INT(D1*X) \ D2=INT(D2*Y) \ P=P1+B-D1-D2
1340 REM**** PRINT RESULTS OF TURN
1350 %\%\%\% "RESULTS OF TURN "#T#" ---"\%
1360 % TAB(5)#"AMOUNT OF FOOD PRESENT IN RANGE---  "#F1#" UNITS"
1370 % TAB(5)#"AMOUNT OF FOOD TAKEN BY COMPETITORS---  "#F2#" UNITS"
1380 % TAB(5)#"AMOUNT OF FOOD FOR THE POPULATION---  "#F3#" UNITS"
1390 % TAB(5)#"DEATHS DUE TO PREDATION---  "#D1
1400 % TAB(5)#"DEATHS DUE TO STARVATION OR OTHER CAUSES---  "#D2
1410 % TAB(5)#"BIRTHS---  "#B
1420 %\% TAB(10)#"CURRENT POPULATION---  "#P \%\%
```

```
1430 REM**** CHECK FOR END OF GAME
1440 IF T=5 THEN 1660
1450 REM**** CHECK FOR LOSS OR GREAT REDUCTION OF THE POPULATION
1460 IF P=0 THEN 1500
1470 IF P>0 AND P<=10 THEN 1530
1480 IF P>10 AND P<=50 THEN 1560
1490 GOTO 1580
1500 % "HEY TURKEY, YOU MANAGED TO LET YOUR POPULATION GO TO ZERO!!"
1510 % "AND, YOU DID IT AFTER ONLY TURN "ǂTǂ" !!!"
1520 GOTO 1670
1530 % "LOOK OUT TURKEY!  YOUR POPULATION IS DOWN TO A MERE"
1540 % Pǂ" INDIVIDUALS!  THAT IS EXTREMELY RISKY!!!"
1550 GOTO 1580
1560 % "YOU HAD BETTER WATCH OUT!  YOUR POPULATION IS DOWN TO"
1570 % Pǂ" INDIVIDUALS!  TROUBLE IS JUST AROUND THE CORNER."
1580 % "TO BEGIN TURN "ǂT+1ǂ" TYPE GO"
1590 INPUT G$
1600 IF G$<>"GO" THEN 1620
1610 GOTO 660
1620 % "IMPROPER RESPONSE--TRY AGAIN." \ GOTO 1580
1660 REM**** FINAL PERFORMANCE EVALUATION
1670 %\% "END OF GAME"\%\%
1680 IF P=0 THEN 1770
1690 IF P>0 AND P<=25 THEN 1800
1700 IF P>25 AND P<=100 THEN 1820
1710 IF P>100 AND P<=250 THEN 1830
1720 IF P>250 AND P<=500 THEN 1840
1730 IF P>500 AND P<=1000 THEN 1850
1740 IF P>1000 AND P<=5000 THEN 1870
1750 IF P>5000 AND P<=10000 THEN 1880
1760 IF P>10000 THEN 1900
1770 % "YOU ARE A DISGRACE!  YOU ARE HEREBY DRUMMED OUT OF THE"
1780 % "AMERICAN COLLEGE OF ECOLOGISTS (ACE) AND MADE A MEMBER"
1790 % "OF THE AMERICAN SOCIETY OF STUPIDOS (YOU KNOW)!!!"\ GOTO 1920
1800 % "THAT'S AWFUL!  THE WILDLIFE MANAGERS OF THE WORLD HAVE"
1810 % "NOTHING BUT SPITE FOR YOU!"\ GOTO 1920
1820 % "THAT'S VERY POOR!  I HOPE YOU DON'T OWN PETS!"\ GOTO 1920
1830 % "THAT'S BARELY PASSABLE!  AT LEAST YOU DIDN'T LOSE ANY!"\ GOTO 1920
1840 % "NOT BAD!  YOU ARE APPOINTED A JUNIOR ECOLOGIST!"\ GOTO 1920
1850 % "PRETTY GOOD JOB FELLA (OR GAL)!  I GUESS WE'LL HAVE TO LET"
1860 % "YOU JOIN THE ECOLOGIST'S CLUB!"\ GOTO 1920
1870 % "GOOD JOB!!  YOU'RE A TRUE BLUE ECOLOGIST!"\ GOTO 1920
1880 % "WOW!!!  TREMENDOUS JOB!  YOU CAN TAKE MARLIN PERKINS' PLACE"
1890 % "ON WILD KINGDOM WHEN HE RETIRES!!"\ GOTO 1920
1900 % "SUPER!!!!  SUPERB!!!!  FANTASTIC!!!!  MOTHER NATURE HERSELF"
1910 % "MUST BE ENVIOUS!!"
1920 %\%\% "DO YOU WANT TO PLAY ANOTHER GAME OF NICHE (Y/N)"ǂ
1930 INPUT Q$
1940 ON ERROR GOTO 1920
1950 IF Q$="Y" OR Q$="YES" THEN 10
1960 GOTO 9999
3000 IF E$="P" THEN E=1
3010 IF E$="L" THEN E=2
3020 IF E$="R" THEN E=3
3030 IF E$="C" OR E$="I" THEN E=4
3040 IF E$="D" OR E$="G" THEN E=5
3050 IF O=1 AND O+E<4 THEN E1=0
3060 IF O=1 AND O+E>=4 THEN E1=1
3070 IF O=2 AND O+E=4 THEN E1=0
3080 IF O=2 AND O+E=3 THEN E1=.33
3090 IF O=2 AND O+E>4 THEN E1=1
3100 IF O=3 AND O+E=6 THEN E1=0
3110 IF O=3 AND O+E<>6 THEN E1=1
3120 IF O=4 AND O+E=8 THEN E1=0
3130 IF O=4 AND O+E=7 THEN E1=.33
3140 IF O=4 AND O+E>8 THEN E1=1
3150 IF O=4 AND O+E<7 THEN E1=1
3160 IF O=5 AND O+E=10 THEN E1=0
3170 IF O=5 AND O+E=9 THEN E1=.5
3180 IF O=5 AND O+E=8 THEN E1=.5
3190 IF O=5 AND O+E<8 THEN E1=1
3200 IF O=3 OR O=4 OR O=5 THEN U$="HECTARES"
3210 IF O=2 THEN U$="DEKASTERES"
3220 IF O=1 THEN U$="CUBIC CENTIMETERS"
3230 RETURN
4000 % TAB(5)ǂ"PARAMECIUM AURELIA (A)"
4010 % TAB(5)ǂ"AMOEBA (B)"
4020 % TAB(5)ǂ"PARAMECIUM BURSARIA (C)"
4030 INPUT C$
4040 ON ERROR GOTO 4070
4060 GOTO 4080
4070 % "IMPROPER RESPONSE--TRY AGAIN."\ GOTO 920
4080 R3=P \ GOTO 980
4100 % TAB(5)ǂ"SNAPPING TURTLE (D)"
4110 % TAB(5)ǂ"CATFISH (E)"
4120 % TAB(5)ǂ"SOCKEYE SALMON (F)"
4130 INPUT C$
4140 ON ERROR GOTO 4170
4160 GOTO 4180
4170 % "IMPROPER RESPONSE--TRY AGAIN."\ GOTO 920
4180 R3=2*100*P\ GOTO 980
4200 % TAB(5)ǂ"PACA (G)"
4210 % TAB(5)ǂ"ARMADILLO (H)"
4220 % TAB(5)ǂ"TAPIR (I)"
4230 INPUT C$
4240 ON ERROR GOTO 4270
4260 GOTO 4280
4270 % "IMPROPER RESPONSE--TRY AGAIN."\ GOTO 920
4280 R3=.6*P \ GOTO 980
4300 % TAB(5)ǂ"CHIPMUNK (J)"
4310 % TAB(5)ǂ"TURKEY (K)"
4320 % TAB(5)ǂ"MOUSE (L)"
4330 INPUT C$
4340 ON ERROR GOTO 4370
4360 GOTO 4380
4370 % "IMPROPER RESPONSE--TRY AGAIN."\ GOTO 920
4380 R3=.05*P \ GOTO 980
4400 % TAB(5)ǂ"PECCARY (M)"
4410 % TAB(5)ǂ"JACKRABBIT (N)"
4420 % TAB(5)ǂ"POCKET MOUSE (O)"
4430 INPUT C$
4440 ON ERROR GOTO 4470
4460 GOTO 4480
4470 % "IMPROPER RESPONSE--TRY AGAIN."\ GOTO 920
4480 R3=.1*P \ GOTO 980
5000 R1=R/R3
5010 IF R1>0 AND R1<=1.33 THEN R2=0
5020 IF R1>1.33 AND R1<=1.67 THEN R2=.05
5030 IF R1>1.67 AND R1<=2 THEN R2=.075
5040 IF R1>2 AND R1<=5 THEN R2=.1
5050 IF R1>5 AND R1<=10 THEN R2=.2
5060 IF R1>10 AND R1<=25 THEN R2=.3
5070 IF R1>25 THEN R2=.4
5080 RETURN
6000 IF F4>1 THEN F4=1
6010 IF F4=0 THEN F5=0
6020 IF F4>0 AND F4<=.25 THEN F5=.1
6030 IF F4>.25 AND F4<=.5 THEN F5=.33
6040 IF F4>.5 AND F4<=.7 THEN F5=.67
6050 IF F4>.7 AND F4<=.9 THEN F5=1
6060 IF F4>.9 AND F4<=1 THEN F5=2
6070 RETURN
9999 END
```

ALL WATCHED OVER BY MACHINES OF LOVING GRACE

I like to think (and
the sooner the better!)
of a cybernetic meadow
where mammals and computers
live together in mutually
programming harmony
like pure water
touching clear sky.

I like to think
(right now, please!)
of a cybernetic forest
filled with pines and electronics
where deer stroll peacefully
past computers
as if they were flowers
with spinning blossoms.

I like to think
(it has to be!)
of a cybernetic ecology
where we are free of our labors
and joined back to nature,
returned to our mammal
brothers and sisters,
and all watched over
by machines of loving grace.

— Richard Brautigan

Sample Run

```
>>>>>>>>>>  NICHE  <<<<<<<<<<

     AN ECOLOGICAL GAME


WELCOME TO THE GAME OF NICHE!
DO YOU NEED INSTRUCTIONS (Y/N) ? Y


INSTRUCTIONS FOR NICHE---

NICHE REFERS TO ALL OF THE ECOLOGICAL VARIABLES WHICH
RELATE TO A GIVEN ORGANISM--ITS HABITAT, LIVING SPACE,
AND ROLE IN THE ECOSYSTEM.  IN THE GAME OF NICHE YOUR JOB
IS TO FIT A SELECTED ORGANISM INTO ITS NICHE.  YOU SELECT
ONE OF 5 ORGANISMS AND RECEIVE A SMALL POPULATION (100
INDIVIDUALS) OF THE ORGANISMS.  YOUR GOAL IN NICHE IS TO
MAXIMIZE THE SIZE OF YOUR POPULATION BY PROVIDING IDEAL
CONDITIONS FOR GROWTH.  AT THE BEGINNING OF THE GAME (TURN 1)
AND AT EACH SUBSEQUENT TURN YOU WILL INPUT THE
ENVIRONMENT, RANGE, AND COMPETITOR FOR YOUR POPULATION.
ONCE DURING THE GAME, YOU MAY INPUT THE PREDATOR KILL OPTION.
THIS WILL REDUCE BUT NOT ELIMINATE THE EFFECTS OF PREDATION
ON YOUR POPULATION FOR THAT TURN ONLY.  AT THE END OF EACH TURN,
THE FOLLOWING INFORMATION WILL BE OUTPUT:  THE AMOUNT OF
FOOD PRESENT IN THE RANGE, THE AMOUNT OF FOOD TAKEN BY
COMPETITORS, THE AMOUNT OF FOOD AVAILABLE TO YOUR POPULATION,
DEATHS DUE TO PREDATION, DEATHS DUE TO STARVATION AND OTHER
CAUSES, BIRTHS, AND THE NEW TOTAL POPULATION.  OF COURSE,
WHAT YOU INPUT WILL AFFECT THE RESULTS WHICH ARE OUTPUT.

TO CONTINUE INSTRUCTIONS, TYPE GO
? GO


FOR INSTANCE, IF THE INPUT RANGE IS TOO SMALL FOR YOUR
POPULATION, THERE WILL BE INSUFFICIENT FOOD (100 UNITS PER
ORGANISM IS OPTIMUM) AND SOME STARVATION WILL RESULT.
IF THE INPUT RANGE IS TOO LARGE, THERE WILL BE MANY PREDATORS
AND DEATHS DUE TO PREDATION WILL BE HIGH.  THE GAME LASTS
FOR 5 TURNS, EACH OF WHICH CORRESPONDS TO ABOUT 1 TO 2
GENERATIONS OF YOUR ORGANISM.  YOU SHOULD RESPOND TO INPUT
PROMPTS BY TYPING WHAT IS IN THE PARENTHESES FOLLOWING YOUR
CHOICE.  YOU MAY INITIATE EACH NEW TURN BY TYPING GO.
GOOD LUCK!! AND, MAY THE BEST ECOLOGIST WIN!!

TO BEGIN THE GAME, TYPE GO
? GO

HERE WE GO!

WHAT ORGANISM---
     PARAMECIUM CAUDATUM (1)
     RAINBOW TROUT (2)
     ANTEATER (3)
     GRAY SQUIRREL (4)
     KANGAROO RAT (5)
? 1

TURN  1

WHAT ENVIRONMENT---
     FRESH WATER POND (P)
     FRESH WATER LAKE (L)
     TROPICAL RAIN FOREST (R)
     BOREAL CONIFEROUS FOREST (C)
     TEMPERATE DECIDUOUS FOREST (T)
     NORTH AMERICAN DESERT (D)
     TEMPERATE GRASSLAND (G)
? P

WHAT RANGE FOR THE WHOLE POPULATION IN CUBIC CENTIMETERS ---
? 50

WHAT COMPETITOR---
     PARAMECIUM AURELIA (A)
     AMOEBA (B)
     PARAMECIUM BURSARIA (C)
? B

PREDATOR KILL (Y/N)---
? N

RESULTS OF TURN  1  ---
     AMOUNT OF FOOD PRESENT IN RANGE---   5000  UNITS
     AMOUNT OF FOOD TAKEN BY COMPETITORS---   1000  UNITS
     AMOUNT OF FOOD FOR THE POPULATION---   4000  UNITS
     DEATHS DUE TO PREDATION---   22
     DEATHS DUE TO STARVATION OR OTHER CAUSES---   60
     BIRTHS---   33

          CURRENT POPULATION---


TO BEGIN TURN  2  TYPE GO
? GO


TURN  2

WHAT ENVIRONMENT---
     FRESH WATER POND (P)
     FRESH WATER LAKE (L)
     TROPICAL RAIN FOREST (R)
     BOREAL CONIFEROUS FOREST (C)
     TEMPERATE DECIDUOUS FOREST (T)
     NORTH AMERICAN DESERT (D)
     TEMPERATE GRASSLAND (G)
? P

WHAT RANGE FOR THE WHOLE POPULATION IN CUBIC CENTIMETERS ---
? 100

WHAT COMPETITOR---
     PARAMECIUM AURELIA (A)
     AMOEBA (B)
     PARAMECIUM BURSARIA (C)
? A

PREDATOR KILL (Y/N)---
? N


RESULTS OF TURN  2  ---

     AMOUNT OF FOOD PRESENT IN RANGE---   10000  UNITS
     AMOUNT OF FOOD TAKEN BY COMPETITORS---   6000  UNITS
     AMOUNT OF FOOD FOR THE POPULATION---   4000  UNITS
     DEATHS DUE TO PREDATION---   8
     DEATHS DUE TO STARVATION OR OTHER CAUSES---   10
     BIRTHS---   51

          CURRENT POPULATION---   84


TO BEGIN TURN  3  TYPE GO
? GO


TURN  3

WHAT ENVIRONMENT---
     FRESH WATER POND (P)
     FRESH WATER LAKE (L)
     TROPICAL RAIN FOREST (R)
     BOREAL CONIFEROUS FOREST (C)
     TEMPERATE DECIDUOUS FOREST (T)
     NORTH AMERICAN DESERT (D)
     TEMPERATE GRASSLAND (G)
? P

WHAT RANGE FOR THE WHOLE POPULATION IN CUBIC CENTIMETERS ---
? 200

WHAT COMPETITOR---
     PARAMECIUM AURELIA (A)
     AMOEBA (B)
     PARAMECIUM BURSARIA (C)
? C
```

```
PREDATOR KILL (Y/N)---
? N

RESULTS OF TURN  3  ---

     AMOUNT OF FOOD PRESENT IN RANGE---   20000  UNITS
     AMOUNT OF FOOD TAKEN BY COMPETITORS---    0  UNITS
     AMOUNT OF FOOD FOR THE POPULATION---   20000  UNITS
     DEATHS DUE TO PREDATION---   28
     DEATHS DUE TO STARVATION OR OTHER CAUSES---   0
     BIRTHS---   168

          CURRENT POPULATION---   224

TO BEGIN TURN  4  TYPE GO
? GO

TURN  4

WHAT ENVIRONMENT---
     FRESH WATER POND (P)
     FRESH WATER LAKE (L)
     TROPICAL RAIN FOREST (R)
     BOREAL CONIFEROUS FOREST (C)
     TEMPERATE DECIDUOUS FOREST (T)
     NORTH AMERICAN DESERT (D)
     TEMPERATE GRASSLAND (G)
? P

WHAT RANGE FOR THE WHOLE POPULATION IN CUBIC CENTIMETERS ---
? 250

WHAT COMPETITOR---
     PARAMECIUM AURELIA (A)
     AMOEBA (B)
     PARAMECIUM BURSARIA (C)
? C

PREDATOR KILL (Y/N)---
? Y

RESULTS OF TURN  4  ---

     AMOUNT OF FOOD PRESENT IN RANGE---   25000  UNITS
     AMOUNT OF FOOD TAKEN BY COMPETITORS---    0  UNITS
     AMOUNT OF FOOD FOR THE POPULATION---   25000  UNITS
     DEATHS DUE TO PREDATION---   4
     DEATHS DUE TO STARVATION OR OTHER CAUSES---   0
     BIRTHS---   448

          CURRENT POPULATION---   668

TO BEGIN TURN  5  TYPE GO
? GO

TURN  5

WHAT ENVIRONMENT---
     FRESH WATER POND (P)
     FRESH WATER LAKE (L)
     TROPICAL RAIN FOREST (R)
     BOREAL CONIFEROUS FOREST (C)
     TEMPERATE DECIDUOUS FOREST (T)
     NORTH AMERICAN DESERT (D)
     TEMPERATE GRASSLAND (G)
? P

WHAT RANGE FOR THE WHOLE POPULATION IN CUBIC CENTIMETERS ---
? 1000

WHAT COMPETITOR---
     PARAMECIUM AURELIA (A)
     AMOEBA (B)
     PARAMECIUM BURSARIA (C)
? C

RESULTS OF TURN  5  ---

     AMOUNT OF FOOD PRESENT IN RANGE---   99999  UNITS
     AMOUNT OF FOOD TAKEN BY COMPETITORS---    0  UNITS
     AMOUNT OF FOOD FOR THE POPULATION---   99999  UNITS
     DEATHS DUE TO PREDATION---   154
     DEATHS DUE TO STARVATION OR OTHER CAUSES---   0
     BIRTHS---   1336

          CURRENT POPULATION---   1850

END OF GAME

GOOD JOB!! YOU'RE A TRUE BLUE ECOLOGIST!

DO YOU WANT TO PLAY ANOTHER GAME OF NICHE (Y/N)? Y

>>>>>>>>>>  NICHE  <<<<<<<<<<

     AN ECOLOGICAL GAME

WELCOME TO THE GAME OF NICHE!
DO YOU NEED INSTRUCTIONS (Y/N) ? N

HERE WE GO!

WHAT ORGANISM---
     PARAMECIUM CAUDATUM (1)
     RAINBOW TROUT (2)
     ANTEATER (3)
     GRAY SQUIRREL (4)
     KANGAROO RAT (5)
? 3

TURN  1

WHAT ENVIRONMENT---
     FRESH WATER POND (P)
     FRESH WATER LAKE (L)
     TROPICAL RAIN FOREST (R)
     BOREAL CONIFEROUS FOREST (C)
     TEMPERATE DECIDUOUS FOREST (T)
     NORTH AMERICAN DESERT (D)
     TEMPERATE GRASSLAND (G)

? L

WHAT RANGE FOR THE WHOLE POPULATION IN HECTARES ---
(1 HECTARE = 10000 SQUARE METERS)
? 1000

WHAT COMPETITOR---
     PACA (G)
     ARMADILLO (H)
     TAPIR (I)
? G

PREDATOR KILL (Y/N)---
? N

RESULTS OF TURN  1  ---

     AMOUNT OF FOOD PRESENT IN RANGE---   166666  UNITS
     AMOUNT OF FOOD TAKEN BY COMPETITORS---    0  UNITS
     AMOUNT OF FOOD FOR THE POPULATION---   166666  UNITS
     DEATHS DUE TO PREDATION---   0
     DEATHS DUE TO STARVATION OR OTHER CAUSES---   100
     BIRTHS---   0

          CURRENT POPULATION---   0

HEY TURKEY, YOU MANAGED TO LET YOUR POPULATION GO TO ZERO!!
AND, YOU DID IT AFTER ONLY TURN  1  !!!

END OF GAME

YOU ARE A DISGRACE!  YOU ARE HEREBY DRUMMED OUT OF THE
AMERICAN COLLEGE OF ECOLOGISTS (ACE) AND MADE A MEMBER
OF THE AMERICAN SOCIETY OF STUPIDOS (YOU KNOW)!!!

DO YOU WANT TO PLAY ANOTHER GAME OF NICHE (Y/N)? N
```

Odell 2: An Elementary Simulation

Marge Kosel and Mike Fish

This simulation allows the user to role play one of four animals: Red Fox, Snowshoe Rabbit, Mouse, or Wolf, and make decisions regarding their survival as they encounter animals such as the owl, hawk, snake, deer, bear, and racoon, and vegetation such as clover, roots, grass, berries, and bark.

Introduction

Many computer simulations have been developed in the State of Minnesota for use by the educational community. The ODELL2 simulation below was derived from a program called ODELL1 created by Ken Witte, a science teacher in Tracy, Minnesota. ODELL1 dealt with food chains among fish in Odell Lake. ODELL1 was used by 10th grade biology students and correlated with a unit from the green version of Houghton-Mifflin biology textbook.

About The Authors

Marge was a former classroom teacher who has been an instructional coordinator for M.E.C.C. the last five years.

Mike will be a senior at Hibbing High School. He has worked for M.E.C.C. developing computer programs and has also done work for independent companies.

Minnesota Educational Computer Consortium, Northeast Instructional Office, Hibbing Community College, Hibbing, MN 55746. (218) 262-3980.

Because of the popularity of the ODELL1 program and because food chains and animal community relationships were covered in lower grade levels, a new simulation called ODELL2 was created. This simulation dealt with the animal relationships.

Instructional Objectives

After the students complete running the program, they should be able to:

1. Describe the characters of each of the four main animals.
2. Described the relationship of each of four main animals to the other animals in the woods.
3. List at least two items which serve as food sources for each animal.
4. If introducted to the concept of herbivores or carnivores, classify the animals as herbivores or carnivores.

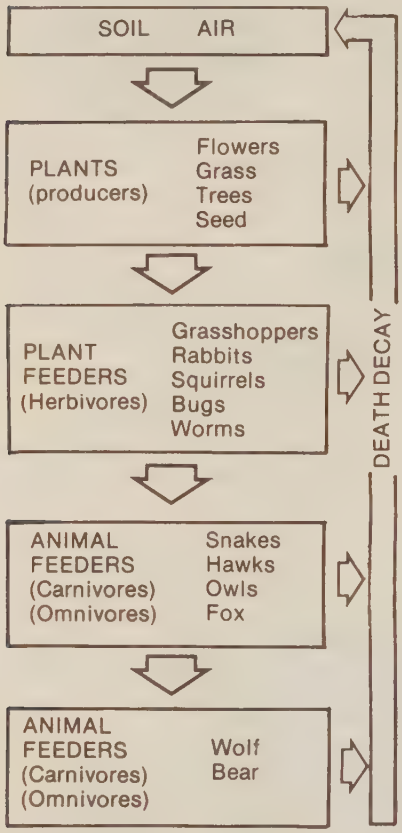
Program Description

The use of this program allows the student to role play animals and actively take part in their community relationships. It provides the students an opportunity to make decisions as they would if they were the animals and see the consequences of those decisions. By providing feedback on each student response, the program will teach the student what the correct responses should be. Minnesota animals were chosen for this simulation, so the students could expand their knowledge of the animals in areas around them.

Background Information

While all plants require sunlight, soil, water, and minerals to grow, different animals require different kinds of food. Some animals live primarily on plants and others eat meat. Regardless of what an animal eats, all food consumed by animals is ultimately derived from plants.

When viewing the food chain there are basically five groups.

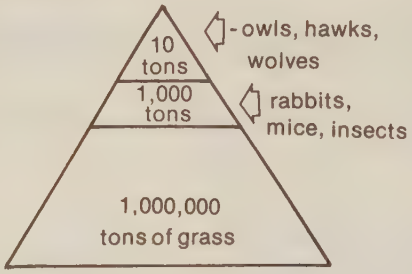


At the bottom of the food chain or cycle are soil, sunlight, and water. Any food cycle is a system of transferring energy. Energy is passed from the sun to the second level - plants. Small animals feed on and derive their energy from the plants. These animals are called herbivores since they feed on herbs and green plants. Carnivorous animals are animals that eat meat instead of just plants, and omnivores are animals that feed on both animals and plants. An animal like a bear is omnivorous. All animals replenish the soil and minerals through decay.

At the top of the food chain, but not an integral part of it, is Man. Man often interferes in the complex cycle provided by nature. As wolves, foxes and animals at the top of the cycle are killed, other parts of the chain may grow out of proportion. For example, if man were to kill the hawks in a certain area, the woods may become overrun with rabbits or mice and this might cause the grass to be completely eaten away and cause erosion or leave little food for other herbivores.

The food cycle has been described as a pyramid of numbers. Animals at the bottom of the pyramid are typically small in size and larger in number. As one moves toward the top of the pyramid, the size of the animals increase while their numbers decrease.

A typical pyramid of numbers for the area is shown in the following diagram:



Looking at the pyramid, you will notice that grass which is at the bottom of the food cycle is most abundant — 1,000,000 tons. About 1,000 tons of plant feeding animals like rabbits, mice, and insects exist while only 10 tons of animals like owls, hawks, and wolves are present. The large animals which occupy the dominant position in the food chain are least in number.

The animals chosen in the simulation are animals common to Minnesota. Below are short abstracts about each of the four main animals.

Timber Wolf

The timber wolf is found in the wilder portions of the north woods. Now fairly abundant in Northeastern Minnesota, the wolf prefers to stay in the woods except when the desire for food brings it into open areas. The wolves feed on a variety of animals from mice to deer including rodents, rabbits, and large birds. Wolves usually travel in packs and their keen sense of smell, vision, and hearing make them excellent hunters.

The adult wolf is about 1.5 meters (5 feet) in length, 1.2 meters tall (4 feet) and weighs from 26-36 kilograms (58-79 pounds). The adult male wolf will occasionally reach weights of 72 kilograms (150 pounds). Wolves have very muscular bodies, long legs, curved teeth and slanted eyes. Their fur is gray in color except for the grayish-white underside.

The wolf's cry is a long monotone sound lacking the yapping of the coyote. Living to be about 16 years old, the wolves usually live in dens in rocky areas. They have litters of 4-12 pups which are blind when they are born and are weaned after about 2 months.

Red Fox

The red fox can be recognized by its reddish gray color and bushy tail. The feet and legs are black in color with the tip of the tail being white. The fox's body is approximately 1 meter (3 feet) long and he has a .6 meter (2 foot) tail and weighs about 7 kilograms (5 pounds). Although the fox usually sleeps in the open, a den, often borrowed from a woodchuck, serves as a nursery when the litter is young. Four to ten pups are born to each litter. When upset, the fox will cry out with a harsh, penetrating bark.

The fox is omnivorous and eats meat such as rabbits and mice as well as berries, fruits, and nuts. On occation foxes will prey upon domestic animals such as poultry.

Noted for being an intelligent, alert, cunning animal, the fox mainly hunts at night.

Snowshoe Rabbit

The snowshoe rabbit's coat changes color seasonally; the summer coat is a dark brown fur and the winter fur changes to snow white. Weighing 1-2 kilograms (2-5 pounds), rabbits are less than ½ meter (1½ feet) long with 13 centimeter (5 inches) hind feet and 7 centimeter (3 inches) ears. Their large feet help them move on the snow. The rabbits have keen eyes to protect them from predatory animals and birds and spend their entire life in a few acres of land.

Rabbits have litters of 2 to 7 furry, open-eyed bunnies and within 2 weeks the little rabbits are eating grasses and summer vegetation.

The snowshoe rabbit grooms itself like a cat and takes dust baths to keep clean. In wooded areas, where the rabbits become numerous, they form well established trails in the snow.

In winter, the rabbits eat twigs, buds, and bark, while in the summer green vegetation is its main meal. The rabbit's most common enemies are fox, badger, man and predatory birds such as hawk, eagle, owl and raven. When being chased, the rabbit usually circles the area and thumps warnings of danger.

Field Mouse

An energetic little creature, the

field mouse eats his own weight every 24 hours. Chewing out a network of paths, clipping grass, eating seed, and tramping the surface with its busy feet, the mouse is a builder of one inch highways. On the move day and night, the mouse will die of exhaustion before it is one year old if it has not been caught and eaten by a predator before that time.

Snakes attack the mouse in its burrow, while hawks, owls, crows, herons, and gulls attack from the air. Bullfrogs, trout, and turtles snap it up when it takes a swim while skunks, foxes, and bobcats feed on the mouse in the open. One weasel can wipe out an entire field of mice and even the great brown bear will dig a mouse out of its burrow for food.

Because mice are prey for so many animals, as the pyramid of number shows, mice are plentiful. A litter will consist of 5-9 young with up to 17 litters being born in one year. The young mice are born in a grass nest inside a burrow or in a tuft of weeds. In three weeks they are out on their own and are full grown by three months.

An adult mouse is 13 to 17 centimeters (5½-7 inches) long and weighs 28-70 grams (1-2½ ounces). The male mouse digs a burrow which has side tunnels leading to toilets and escape routes. Some of the tunnels made by mice have been measured to be 150 meters (350 feet) long. The females seldom venture further than 10 meters (20-30 feet) from their birthplace. The mouse, like a squirrel, stores berries, bulbs, roots and seed along its tunnels for the winter.

Description of the ODELL2 Program and Model

The ODELL2 program allows the user to role play one of the following animals:

Field Mouse
Snowshoe Rabbit
Red Fox
Timber Wolf

Each animal will randomly encounter one of ten situations. If the students survive the ten situations it is assumed that they understand the role of that animal in the food chain. The students can make one of four decisions for their animals when they encounter a situation:

1. Run from the situation.
2. Ignore the situation or problem
3. Attempt to eat the animal or vegetation
4. Chase it out of their territory

The program also keeps track of the animals injuries and the energy it uses and acquires through food. If the animal's energy level gets too low, it will die of starvation. If its injury level is high and it encounters an enemy, its chances of being caught are higher even though students have selected a correct move, such as running from the situation. The model, in order to make the simulation more realistic, is designed so that even if the animal makes the right decision, the outcome is not always what the animal would like. For example, if the wolf attacked the fox, there is a 20% chance that the fox may escape the wolf's attack. The sample chart shows situations the rabbit will encounter and the consequences of the different actions.

Program Listing

A listing of ODELL2 is available for $1.00 postpaid from Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

Information on MECC software is contained in the booklet "MECC Program Distribution Policy" available for $1.00 postpaid from MECC Publications, 2520 Broadway Drive, Lauderdale, MN 55113. □

```
WOULD YOU LIKE TO SEE THE INSTRUCTIONS?  YES

     ODELL2 IS A PROGRAM WHICH SIMULATES
THE LIFE OF ANIMALS IN THE WOODS WHICH
ARE AROUND ODELL LAKE.  YOU PLAY THE ROLE
OF AN ANIMAL IN THE WOODS.
     THERE ARE SEVERAL DIFFERENT ANIMALS
FROM WHICH YOU CAN CHOOSE.  THEY ARE--

          SNOWSHOE RABBIT
            FIELD MOUSE
              RED FOX
             TIMBER WOLF

WHEN ASKED, TYPE IN THE NAME OF THE ANIMAL
YOU WOULD LIKE TO BE.
     THERE ARE A FEW THINGS THAT DECIDE WHETHER
OR NOT YOU MAKE IT THROUGH THE WOODS--

       1.  HOW WELL YOU SURVIVE AN ATTACK.
       2.  YOUR HUNGER LEVEL
       3.  YOUR INJURY LEVEL

WHICH ANIMAL WOULD YOU LIKE TO BE?RABBIT

WOULD YOU LIKE TO HAVE THE OPTIONS FOR A RABBIT?YES
THERE ARE SEVERAL OPTIONS THAT YOU
CAN USE WHILE GOING THROUGH THE ODELL
WOODS.  FOR A RABBIT THEY ARE--
       1.  HOP AWAY WITH AN ATTEMPT TO ESCAPE
       2.  IGNORE THE PROBLEM ALL TOGETHER
       3.  STOP AND EAT IT
       4.  CHASE IT OUT OF YOUR TERRITORY
 TYPE A 1, 2, 3, OR 4.?

ON YOUR TRAVELS THROUGH THE WOODS,
YOU CAME ALMOST A RACCOON.
WHAT WILL YOU DO?1
THAT WAS SMART, BECAUSE RACCOONS ARE MORE FEROCIOUS
THAN MOST PEOPLE THINK.

 AS YOU ARE HOPPING DOWN THE PATH, YOU
 COME UP TO AN OLD FARMER'S HUNTING DOG.
 WHAT WILL YOU DO?1
 YOU WERE SAFE THIS TIME.  THE
 DOG WAS TOO OLD.

AS YOU WALK DOWN THE PATH, YOU NOTICE A MOUSE OFF
TO THE SIDE.  WHAT WILL YOU DO?3
YOU WOULD NOT EAT A MOUSE.

AS YOU GO DOWN THE TRAIL FARTHER, YOU COME
UP TO A DEER.  WHAT WILL YOU DO?2
GOOD---A DEEP WOULDN'T HURT YOU.

THERE IS AN OWL SITTING ON A BRANCH
IN THE TREE BEHIND YOU.
WHAT DO YOU DO?1
THAT WAS SMART OF YOU
TO ESCAPE FROM THE OWL -- IT HAD ITS
EYE ON YOU.

OFF THE PATH
IS A LARGE CLOVER FIELD.
WHAT ARE YOU GOING TO DO?3
GOOD IDEA...THAT SURE IS GOOD CLOVER.
NOW THAT YOU ARE DONE EATING, LET'S MOVE ON...

AS YOU ARE GOING DOWN THE PATH,
YOU COME ACROSS A RED FOX.
WHAT ARE YOU GOING TO DO?1
YOU WERE VERY SMART
TO ESCAPE.  YOU ARE SAFE NOW, BUT YOUR ENERGY
WENT DOWN FROM THE RUNNING.

WALKING DOWN THE TRAIL,
YOU COME TO A HUNGRY WOLF PACK.
WHAT ARE YOU GOING TO DO?1
THAT WAS A GOOD CHOICE TO
RUN FROM THE WOLF PACK,
THEY HARDLY NOTICED YOU.
BUT YOU DID LOSE SOME ENERGY
BY RUNNING SO FAST.

 THERE IS A FALLEN TREE IN FRONT OF YOU.
 WHAT ARE YOU GOING TO DO?3
 GOOD!!!THAT BARK SURE TASTES GOOD TO A RABBIT
 DOESN'T IT?  WELL...WE'D BETTER MOVE ON...

 THERE IS A GRASSY FIELD OFF TO YOUR SIDE...
 WHAT ARE YOU GOING TO DO?3
 GOOD...A GRASSY FIELD IS GOOD FOR A RABBIT.
 NOW THAT YOU'VE HAD A FULL MEAL, IT IS TIME TO MOVE ON.

 ************************************

 CONGRATULATIONS...YOU MADE IT THROUGH ODELL WOODS
 WITH NO INJURIES.
    I HOPE THAT YOU HAD FUN.  COME BACK AGAIN
 AND TRY TO MAKE IT THROUGH THE ODELL WOODS.
```

```
1  TEXT : HOME : PRINT : PRINT : PRINT
2  PRINT  TAB( 17);"ODELL2"
3  FOR I = 1 TO 1000: NEXT I: HOME
4  TEXT : HOME : PRINT "THIS PROGRAM IS ABOUT THE ANIMALS THAT": PRINT
5  PRINT "LIVE AROUND ODELL LAKE.  YOU WILL ROLE": PRINT
6  PRINT "A ANIMAL."
7  PRINT
8  PRINT "THERE ARE A FEW THINGS THAT DECIDE": PRINT
9  PRINT "WHETHER YOU MAKE IT THROUGH THE WOODS"
10  PRINT : PRINT " 1.  HOW WELL YOU SURVIVE ATTACK": PRINT " 2.  YOUR HUNGER LEVEL": PRINT " 3.  YOUR INJURY LEVEL"
11  VTAB (23): PRINT "    PRESS 'RETURN' TO CONTINUE"
12  GET A$
40  CLEAR :J$ = "WHAT WOULD YOU LIKE TO DO"
50  DIM H(11):I = 0:E = 46
100  POKE 34,0: TEXT : HOME : PRINT
150  PRINT : PRINT "YOU CAN BE:"
151  PRINT "     RABBIT": PRINT "     MOUSE": PRINT "     FOX": PRINT "     WOLF": PRINT
180  PRINT "WHICH ANIMAL WOULD YOU LIKE TO BE";: INPUT F$
190  PRINT
200  IF F$ <  > "RABBIT" THEN 00220
210 A = 1:E = E + 15: GOTO 300
220  IF F$ <  > "MOUSE" THEN 00240
230 A = 2:E = E + 10: GOTO 300
240  IF F$ <  > "FOX" THEN 00260
250 A = 3: GOTO 300
260  IF F$ <  > "WOLF" THEN 00280
270 A = 4:E = E - 6: GOTO 300
280  PRINT : PRINT "PLEASE BE A RABBIT,MOUSE,": PRINT "FOX, OR A WOLF.": PRINT : GOTO 180
300  TEXT : HOME : PRINT : PRINT "AS A ";F$;" YOUR OPTIONS ARE:"
301  PRINT  TAB( 10);"1. RUN": PRINT  TAB( 10);"2. IGNORE IT": PRINT  TAB( 10);"3. EAT IT": PRINT  TAB( 10);"4. CHASE IT"
302  POKE 34,8
310  PRINT : PRINT
320  FOR Y = 1 TO 10
330 X =  INT ( RND (1) * 10) + 1
340  FOR N = 1 TO Y
350  IF H(N) = X THEN 00330
360  NEXT N
380 H(Y) = X
400  ON X GOSUB 00580,01380,02390,03000,03370,03970,04750,05070,05830,06500
410  GOSUB 07680
420  NEXT Y
430  PRINT
440  POKE 34,0: HOME : PRINT : PRINT : PRINT "CONGRATULATIONS...": PRINT "YOU MADE IT THROUGH ODELL WOODS "
450  PRINT "WITH ";
460  IF I < 0 THEN  PRINT "NO ";: GOTO 530
470  IF I < 10 THEN  PRINT "ALMOST NO ";: GOTO 530
480  IF I < 30 THEN  PRINT "SOME ";: GOTO 530
490  PRINT "HEAVY ";
530  PRINT "INJURIES."
540  PRINT : PRINT "I HOPE YOU HAD FUN.": PRINT : PRINT : PRINT "COME BACK AGAIN"
570  GOTO 07950
580  ON A GOTO 00590,00590,00920,00920
590  REM
600  PRINT : PRINT "AS YOU ARE GOING DOWN THE PATH,"
610  PRINT "YOU COME ACROSS A RED FOX."
620  PRINT : PRINT J$;: INPUT V: PRINT : HOME
630  IF V < 1 OR V > 4 THEN 620
660  ON V GOTO 670,790,870,890
670 R =  INT ( RND (1) * 100) + 1 - I
680  IF R <  = 10 THEN 00770
690  IF R <  = 20 THEN 00740
700  PRINT "YOU WERE VERY SMART TO ESACPE."
710  PRINT "YOU ARE SAFE NOW, BUT YOUR ENERGY"
720  PRINT "WENT DOWN FROM RUNNING."
730 E = E - 10: RETURN
740  PRINT "AS YOU WERE RUNNING AWAY, THE FOX "
750  PRINT "INJURED YOU WITH HIS SHARP TEETH."
760 I = I + 10: RETURN
770  PRINT "THE FOX CAUGHT UP AND GOT YOU."
780 B$ = "FOX": GOTO 8010
790 R =  INT ( RND (1) * 100) + 1
800  IF R < 90 THEN 00850
810  PRINT "YOU WERE LUCKY, THE FOX"
820  PRINT "DIDN'T NOTICE YOU THERE. YOU"
830  PRINT "SHOULD BE MORE CAREFUL."
840  RETURN
850  PRINT "THAT WASN'T A SMART IDEA..."
860  GOTO 00780
870  PRINT "YOU GOT KILLED TRYING."
880  GOTO 00780
890  PRINT "THE FOX FOUND YOU"
900  PRINT "AND CAUGHT YOU."
910  GOTO 00780
920  PRINT : PRINT "YOU SEE A RABBIT IN ";
930  PRINT "YOUR TRAVELS."
940  PRINT : PRINT J$;: INPUT V: PRINT : HOME
950  IF V < 1 OR V > 4 THEN 940
980  ON V GOTO 990,1030,1070,1160
985  RETURN
990  PRINT "THERE IS NO REASON TO TRY TO RUN"
1000  PRINT "FROM A RABBIT. REMEMBER..."
1005  PRINT "YOU ARE A ";F$;",AND YOU JUST"
1010  PRINT "GOT WEAKER FROM RUNNING FROM A RABBIT."
1020 E = E - 15: RETURN
1030  PRINT "OF COURSE "
1040  PRINT "YOU ARE SAFE; BUT YOU SHOULD HAVE EATEN"
1050  PRINT "FOR MORE  ENERGY. "
1060 E = E - 5: RETURN
1070  PRINT "YOU GOT IT  THAT WAS A GOOD MOVE"
1075  PRINT "BECAUSE YOU'LL HAVE MORE ENERGY";
1080  IF I = 0 THEN 01110
1090  PRINT "AND YOUR WOUNDS "
1101  PRINT "WILL HEAL FASTER";
1110  PRINT "."
1120 E = E + 15:I = I - 5
1130  PRINT "NOW THAT YOU HAVE FINISHED YOUR MEAL"
1140  PRINT "YOU CAN MOVE ON."
1150  RETURN
1160  PRINT "WHY WOULD YOU CHASE A RABBIT": PRINT "AND NOT EAT IT?": PRINT
1170  PRINT "THE RABRIT RAN INTO THE BUSHES"
1180 E = E - 15
1200  PRINT "AND YOU LOST ENERGY CHASING IT."
1210 R =  INT ( RND (1) * 100) + 1
1220  IF R <  = 20 THEN 01240
1230  RETURN
1240  PRINT "AS YOU CHASED THE RABBIT,"
1270  PRINT "YOU STUMBLED INTO A TRAP."
1280  PRINT "EVEN THOUGH YOU ESCAPE, YOU HAD"
1290  PRINT "MAJOR INJURIES."
1300 I = I + 15: RETURN
1380  PRINT
1390  ON A GOTO 1400,1400,1700,2080
1400  PRINT : PRINT "WALKING DOWN THE TRAIL,"
1410  PRINT "YOU COME TO A HUNGRY WOLF PACK."
1420  PRINT : PRINT J$;: INPUT V: PRINT : HOME
1430  IF V < 1 OR V > 4 THEN 1420
1460  IF V = 1 THEN 01510
1470  PRINT "THAT WASN'T A SMART MOVE ON YOUR PART."
1480  PRINT "THE WOLF PACK ATE YOU."
1490 B$ = "WOLF PACK": GOTO 8010
1500  PRINT
1510  PRINT "THAT WAS A GOOD CHOICE TO"
1520  PRINT "RUN FROM THE WOLF PACK,"
1530 R =  INT ( RND (1) * 100) + 1 + I
1540  IF R >  = 95 THEN 01670
```

```
1550  IF R >  = 80 THEN 01600
1560  PRINT "THEY HARDLY NOTICED YOU. "
1570  PRINT "BUT YOU DID LOSE SOME ENERGY"
1580  PRINT "BY RUNNING SO FAST."
1590 E = E - 15: RETURN
1600  PRINT "BUT AS YOU WERE RUNNING YOU LOST ENERGY"
1610  PRINT "AND THE WOLF PACK INJURED YOU ";
1620  IF I > 0 THEN 01650
1630  PRINT "."
1640  GOTO 01660
1650  PRINT "MORE."
1660 I = I + 15:E = E - 15: RETURN
1670  PRINT "BUT UNFORTUNATELY,THE WOLF PACK CAUGHT"
1680  PRINT "YOU AND KILLED YOU."
1690  GOTO 1490
1700  PRINT : PRINT "YOU STROLLED OFF THE PATH AND"
1703  PRINT "SEE A WOLF PACK."
1710 R =  INT ( RND (1) * 2) + 1
1720  IF R = 2 THEN 01730
1730  PRINT "OFF TO YOUR LEFT."
1740  GOTO 01760
1750  PRINT "OFF TO YOUR RIGHT."
1760  PRINT : PRINT J$;: INPUT V: PRINT : HOME
1770  IF V < 1 OR V > 4 THEN 1760
1800  PRINT
1810  ON V GOTO 01820,01970,01970,01970
1820  PRINT "THAT WAS A GOOD IDEA, YOU MUST HAVE"
1830  PRINT "KNOWN THAT A WOLF WILL EAT A FOX."
1840  PRINT "AS YOU RAN FROM THE WOLVES,": PRINT "YOU LOST ENERGY AND"
1850  PRINT "THEY BEGAN TO CATCH UP TO YOU."
1860 E = E - 15
1870 R =  INT ( RND (1) * 100) + 1 + I
1880  IF R >  = 95 THEN 01940
1890  IF R >  = 85 THEN 01920
1900  PRINT "YOU WERE SAFE,": PRINT "BECAUSE YOU RAN LONG AND HARD"
1910  RETURN
1920  PRINT "YOU RAN FAST,": PRINT "BUT THE WOLF PACK INJURED YOU."
1930 I = I + 10: RETURN
1940  PRINT "THE WOLF PACK MOVED IN,": PRINT "AND YOU FELL INTO SOME
1950  PRINT "THEY CAUGHT UP TO YOU AND ATE YOU."                ROCKS."
1960  GOTO 01490
1970  PRINT "THAT WAS A BAD IDEA TO TRY TO"
1980  ON V GOTO 01990,01990,02020,02020
1990  PRINT "IGNORE THE WOLF PACK,": PRINT "BECAUSE IT WAS OUTLOOKING"
2000  PRINT "FOR FOOD, AND YOU WERE IT"
2010  GOTO 01490
2020  PRINT "EAT A WOLF PACK?": PRINT "THERE IS NO WAY YOU COULD"
2030  PRINT "HAVE EATEN A WOLF."
2040  GOTO 0149
2050  PRINT "CHASE A WOLF PACK? YOU LET THEM KNOW "
2060  PRINT "WHERE YOU WERE, AND THEY GOT YOU"
2070  GOTO 01490
2080  PRINT : PRINT "YOU ARE RESTING UNDER A TREE"
2085  PRINT "WHEN YOU SEE A GOPHER."
2090  PRINT : PRINT J$;: INPUT V: PRINT : HOME
2100  IF V < 1 OR V > 4 THEN 2090
2130  ON V GOTO 02140,02180,02210,02350
2140  PRINT "THAT WASN'T A GOOD IDEA TO "
2150  PRINT "ESCAPE FROM A GOPHER. ALL YOU DID WAS"
2160  PRINT "USE UP YOUR ENERGY."
2170 E = E - 15: RETURN
2180  PRINT "OF COURSE YOU CAN IGNORE IT,"
2190  PRINT "BUT THAT WAS ENERGY THAT": PRINT "YOU COULD HAVE USED."
2200  RETURN
2210  PRINT "THAT WAS A GOOD CHOICE, ";
2220  IF E < 30 THEN 02260
2230  PRINT "IT'S ALWAYS GOOD TO BUILD"
2240  PRINT "UP YOUR ENERGY,"
2250  GOTO 02270
2260  PRINT "YOU NEEDED THAT ENERGY,"
2270 R =  INT ( RND (1) * 100) + 1 + I
2280  IF R >  = 95 THEN 02330
2290  IF R >  = 80 THEN 01240
2300  PRINT "AND YOU GOT IT."
2310  PRINT "ALTHOUGH HE DID PUT UP A GOOD STRUGGLE."
2320 E = E + 10:I = I - 10: RETURN
2330  PRINT "BUT HE BURROWED INTO THE GROUND."
2340 E = E - 5: RETURN
2350  PRINT "AS YOU CHASED THE GOPHER INTO THE WOODS,"
2360  PRINT "YOU LOST HIM. BOY, YOU MUST"
2370  PRINT "REALLY FEEL TIRED AFTER THAT RUN"
2380 E = E - 10: RETURN
2390  ON A GOTO 02400,02580,02750,02750
2400  PRINT : PRINT "JUST OFF THE PATH IS A CLOVER FIELD."
2420  PRINT : PRINT J$;: INPUT V: PRINT : HOME
2430  IF V < 1 OR V > 4 THEN 2420
2460  ON V GOTO 02470,02490,02520,02550
2470  PRINT "YOU DON T NEED TO RUN FROM A FIELD."
2480  RETURN
2490  PRINT "YOU SHOULDN'T HAVE IGNORED SUCH A
2500  PRINT "BEAUTIFUL FIELD AS THAT."
2510 E = E - 10: RETURN
2520  PRINT "GOOD IDEA...THAT SURE IS GOOD CLOVER."
2530  PRINT "NOW THAT YOU ARE DONE EATING,": PRINT "LET'S MOVE ON..."
2540 E = E + 15:I = I - 15: RETURN
2550  PRINT "DON'T BE FOOLISH. LET'S MOVE ON."
2560 E = E - 15: RETURN
2570 E = E - 15
2580  PRINT : PRINT "THERE IS A RABBIT IN FRONT OF YOU."
2590  PRINT : PRINT J$;: INPUT V: PRINT : HOME
2600  IF V < 1 OR V > 4 THEN 2590
2630  ON V GOTO 02640,02660,02690,02720
2640  PRINT "SURE, BUT A RABBIT WOULDN'T HURT YOU ."
2650 E = E - 15: RETURN
2660  PRINT "GOOD... IT CAN'T HURT YOU..."
2670  PRINT
2680  RETURN
2690  PRINT "WHY WOULD YOU WANT TO EAT A RABBIT?"
2700  PRINT "IT RAN AWAY BEFORE YOU GOT CLOSE."
2710 E = E - 10: RETURN
2720  PRINT "CHASING A RABBIT WASN'T": PRINT "A VERY GOOD IDEA..."
2730  PRINT "IT HOPPED AWAY, AND YOU LOST ENERGY."
2740 E = E - 15: RETURN
2750  PRINT : PRINT "AS YOU LOOK DOWN THE PATH, YOU SEE"
2760  PRINT "A MOUSE IN FRONT OF YOU..."
2770  PRINT : PRINT J$;: INPUT V: PRINT : HOME
2780  IF V < 1 OR V > 4 THEN 2770
2810  ON V GOTO 02820,02850,02930,02880
2820  PRINT "THINK ABOUT IT...YOU ARE A ";F$;"."
2830  PRINT "YOU CAN SURELY EAT A SMALL MOUSE."
2840 E = E - 15: RETURN
2850  PRINT "YOU COULD HAVE GOTTEN "
2860  PRINT "A MOUSE IF YOU WOULD HAVE TRIED..."
2870 E = E - 5: RETURN
2880  PRINT "THE MOUSE HID UNDER A ROCK "
2890  PRINT "WHERE YOU COULDN'T GET HIM."
2900  PRINT "HE SURE WAS SURPRISED": PRINT "WHEN A ";F$;" TRIED"
2910  PRINT "TO CHASE HIM."
2920 E = E - 15: RETURN
2930 R =  INT ( RND (1) * 100) - 1 + I
2940  IF R >  = 95 THEN 02980
2950  PRINT "YOU GOT IT..."
2960  PRINT "YOU MUST FEEL BETTER ABOUT EATING SOMETHING."
2970 I = I - 10:E = E + 10: RETURN
2980  PRINT "OOPS HE GOT AWAY. HE WAS VERY LUCKY. "
2990 E = E - 10: RETURN
3000  PRINT : PRINT "THERE IS A FALLEN TREE WITH BARK ON IT "
3001  PRINT "IN FRONT OF YOU."
3010  PRINT : PRINT J$;: INPUT V: PRINT : HOME
3020  ON A GOTO 03060,03060,03190,3190
3025  IF V < 1 OR V > 4 THEN 3010
3060  ON V GOTO 3070,3110,3150,3170
3070  PRINT "WHY RUN FROM A TREE? BEING"
3080  PRINT "A ";F$;" YOU WOULD HAVE LIKED TO EAT"
3090  PRINT "BARK FROM THAT TREE..."
3100 E = E - 15: RETURN
3110  PRINT "THE BARK OF A TREE IS GOOD FOOD."
3120  PRINT "BEING A ";F$: PRINT "YOU COULD HAVE BUILT UP YOUR ENERGY."
3140 E = E - 5: RETURN
3150  PRINT "THAT TASTES GOOD.  LETS MOVE ON.."
3160 I = I - 15:E = E + 20: RETURN
3170  PRINT "BUT YOU DID GET HUNGRY FROM TRYING."
3180 E = E   15: RETURN
3190  ON V GOTO 3230,3260,3290,3330
3230  PRINT "BEING A ";F$: PRINT "YOU DON'T NEED TO RUN FROM A TREE."
3240  PRINT "SAVE YOUR ENERGY"
3250 E = E - 15: RETURN
3260  PRINT "GOOD, THE TREE WON'T HELP YOU"
3270  PRINT "MUCH ANYWAY. LET S MOVE ON..."
3280  RETURN
3290  PRINT "EATING BARK FROM TREES IS NOT GOOD FOR"
3300  PRINT "YOU. YOU SHOULD BE MORE CAREFUL": PRINT "ABOUT WHAT YOU EAT."
3320 I = I + 5: RETURN
3330  PRINT "A ";F$;" DOESN T USUALLY GO"
3340  PRINT "AROUND CHASING TREES. WHY CHASE A TREE?"
3350  PRINT "LET'S GO ON..."
3360 E = E - 15: RETURN
3370  PRINT : PRINT "THERE IS AN OWL SITTING ON A BRANCH"
3380  PRINT "IN THE TREE BEHIND YOU."
3390  PRINT : PRINT J$;: INPUT V: PRINT : HOME
3410  IF V < 1 OR V > 4 THEN 3390
3415  ON A GOTO 3440,3440,3710,3710
3440  ON V GOTO 03470,03600,03680,03680
3470 R =  INT ( RND (1) * 100) + 1 + I
3480 E = E - 15
3490  IF R >  = 80 THEN 03540
3500  PRINT "THAT WAS SMART OF YOU"
3510  PRINT "TO ESCAPE FROM THE OWL -- IT HAD ITS"
3520  PRINT "EYE ON YOU."
3530  RETURN
3540  PRINT "AS YOU WERE ESCAPING, THE OWL"
3550  IF R >  = 90 THEN 03580
3560  PRINT "SWOOPED DOWN AND CLAWED YOU LIGHTLY."
3570 I = I + 1: RETURN
3580  PRINT "SWOOPED DOWN AND GOT YOU."
3590 B$ = "OWL": GOTO 8010
```

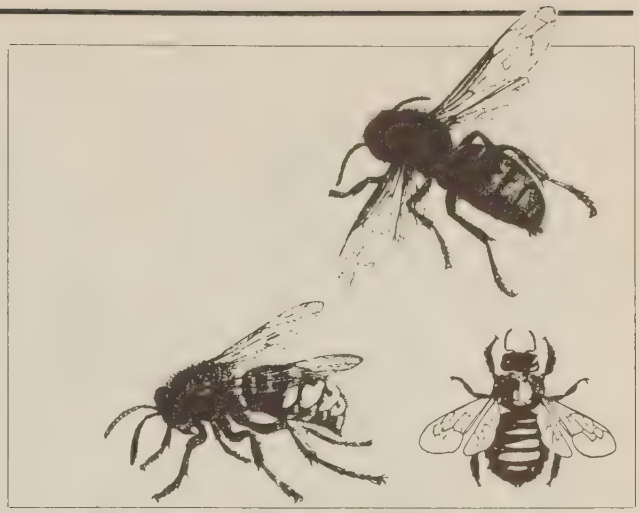
```
3600 R =  INT ( RND (1) * 100) + 1 + I
3610  IF R >  = 40 THEN 03650
3620  PRINT "YOU WERE  LUCKY THAT THE OWL"
3630  PRINT "DIDN'T SEE YOU."
3640  RETURN
3650  PRINT "YOU SHOULDN'T HAVE IGNORED THE OWL."
3660  PRINT "IT FLEW DOWN AND GOT YOU."
3670  GOTO 03590
3680  PRINT "NO, NO, NO...A ";F$;" CAN'T FLY."
3690  PRINT "YOU WERE JUST KILLED BY THE OWL TRYING"
3700  GOTO 03590
3710  ON V GOTO 3760,3790,3810,3940
3760  PRINT "YOU DON'T NEED TO DO THAT---"
3770  PRINT "YOU JUST WASTED ENERGY."
3780 E = E - 15: RETURN
3790  PRINT "O.K. LET'S MOVE ON..."
3800  RETURN
3810  IF A = 3 THEN 03900
3820 R =  INT ( RND (1) * 100) - 1 + I
3830  IF R >  = 50 THEN 03870
3840  PRINT "ALREADY INJURED AND YOU "
3850  PRINT "GOT IT FROM A LOW BRANCH."
3860 E = E + 15:I = I - 15: RETURN
3870  PRINT "NO. WHEN YOU TRIED FOR THE"
3880  PRINT "OWL, IT FLEW FROM ITS BRANCH."
3890 E = E - 15: RETURN
3900  PRINT "YOU WERE JUST INJURED"
3910  PRINT "BY THE OWL AS IT FLEW"
3920  PRINT "PAST YOU ON IT'S ESCAPE."
3930 I = I + 15: RETURN
3940  PRINT "AS YOU RAN UP TO THE OWL'S TREE,"
3950  PRINT "IT FLEW AWAY."
3955 E = E - 15
3960  GOTO 03890
3970  PRINT : ON A GOTO 3980,4310,4530,4530
3980  PRINT "AS YOU ARE HOPPING DOWN THE PATH, YOU"
3990  PRINT "COME UP TO AN OLD FARMER'S HUNTING DOG."
4000  PRINT : PRINT J$;: INPUT V: PRINT : HOME
4010  IF V < 1 OR V > 4 THEN 4000
4040  ON V GOTO 04050,04160,04250,04280
4050 R =  INT ( RND (1) * 100) + 1 + I
4060  IF R >  = 95 THEN 04110
4070  IF R >  = 85 THEN 04130
4080  PRINT "YOU WERE SAFE THIS TIME."
4090  PRINT "THE DOG WAS TOO OLD."
4100 E = E - 10: RETURN
4110  PRINT "YOU GOT AWAY, BUT THE DOG BIT YOU."
4120  GOTO 02540
4130  PRINT "IT WAS RIGHT TO RUN BUT"
4140  PRINT "THE DOG GOT YOU ANYWAY."
4150 B$ = "HUNTING DOG": GOTO 8010
4160  PRINT "REMEMBER...THE DOG IS TRAINED FOR"
4170  PRINT "HUNTING, ";
4180 R =  INT ( RND (1) * 2) - 1
4190  IF R = 2 THEN 04230
4200  PRINT "BUT YOU WERE LUCKY BECAUSE"
4210  PRINT "IT DIDN'T NOTICE YOU."
4220  RETURN
4230  PRINT "AND HE SNIFFED YOU OUT AND CAUGHT YOU"
4240  GOTO 04150
4250  PRINT "WHAT?? YOU'RE TRYING TO EAT A DOG?"
4260  PRINT "YOU GOT KILLED TRYING."
4270  GOTO 04150
4280  PRINT "A RABBIT CAN'T CHASE A DOG---YOU"
4290  PRINT "GOT KILLED TRYING."
4300  GOTO 04150
4310  PRINT "GOING DOWN THE PATH, YOU COME"
4320  PRINT "ACROSS A SNAKE."
4330  PRINT : PRINT J$;: INPUT V: PRINT : HOME
4340  IF V < 1 OR V > 4 THEN 4330
4370  ON V GOTO 04380,04420,04450,04470
4380 R =  INT ( RND (1) * 100) + 1 + I
4390  IF R >  = 90 THEN 04500
4400  PRINT "YOU ARE SAFE. HE DIDN'T NOTICE YOU."
4410 E = E - 15: RETURN
4420  PRINT "MAYBE YOU IGNORED IT, BUT IT"
4440 B$ = "SNAKE": GOTO 8010
4450  PRINT "NO. YOU DIDN'T EAT IT---IT ATE YOU."
4460  GOTO 04440
4470  PRINT "THE SNAKE ATTACKED YOU "
4480  PRINT "AS YOU GOT TOO CLOSE,"
4490  GOTO 04510
4500  PRINT "YOU WEREN'T LUCKY ENOUGH THIS TIME---"
4510  PRINT "IT GOT YOU."
4520  GOTO 04440
4530  PRINT "AS YOU GO DOWN THE PATH, YOU"
4540  PRINT "GO THROUGH A FARMER'S LAND."
4550  PRINT "THE FARMER IS OUT HUNTING."
4560  PRINT : PRINT J$;: INPUT V: PRINT : HOME
4570  IF V < 1 OR V > 4 THEN 4560
4600  PRINT
4610  ON V GOTO 04620,04730,04730,04730
4620 R =  INT ( RND (1) * 100) + 1 + I
4630  IF R >  = 90 THEN 04710
4640  IF R > 80 THEN 04680
4650  PRINT "WHEW---THAT WAS CLOSE; HE"
4660  PRINT "MISSED TWICE."
4670 E = E - 15: RETURN
4680  PRINT "MISSED ONCE--JUST NICKED YOU"
4690  PRINT "WITH THE SECOND BULLET."
4700 E = E - 10:I = I + 15: RETURN
4710  PRINT "HE GOT YOU..."
4720 B$ = "FARMER": GOTO 8010
4730  PRINT "HE GOT YOU."
4740  GOTO 04720
4750  PRINT : PRINT "AS YOU GO DOWN THE TRAIL FARTHER, "
4765  PRINT "YOU COME UP TO A DEER."
4768  PRINT : PRINT J$;: INPUT V: PRINT : HOME
4770  IF V < 1 OR V > 4 THEN 4768
4800  ON A GOTO 04810,04810,04810,04960
4810  ON V GOTO 04820,04860,04880,04880
4820  PRINT "WHY WOULD A ";F$;" ESCAPE FROM A "
4830  PRINT "DEER? A DEER WOULDN'T EAT YOU."
4840  PRINT "LET'S GO ON..."
4850 E = E - 15: RETURN
4860  PRINT "GOOD---A DEER WOULDN'T HURT YOU.
4870  RETURN
4880  GOSUB 04930
4890  PRINT "WHAT?--A ";F$;" DOESN'T ";G$;" A DEER."
4900  PRINT "YOU GOT IT EXCITED AND IT ";
4910  PRINT "RAN AWAY."
4920  RETURN
4930  ON V GOTO 04940,04940,04940,04950
4940 G$ = "EAT":E = E - 15: RETURN
4950 G$ = "CHASE":E = E - 15: RETURN
4960  ON V GOTO 04820,04970,05000,04970
4970  PRINT "YOU SHOULD HAVE TRIED TO EAT THE DEER."
4980  PRINT "YOU HAD A CHANCE OF GETTING IT."
4990 E = E - 5: RETURN
5000 R =  INT ( RND (1) * 100) + 1 + I
5010  IF I < 50 THEN 05040
5020  PRINT "IT WAS CLOSE, BUT THE DEER WAS TOO FAST."
5030 E = E - 15: RETURN
5040  PRINT "WHAT A BIG MEAL.  YOU SHOULDN'T"
5050  PRINT "BE HUNGRY FOR A WHILE---"
5060 E = E + 20:I = I - 15: RETURN
5070  PRINT : ON A GOTO 5080,5240,5400,5570
5080  PRINT "AS YOU WALK DOWN THE PATH, YOU NOTICE"
5081  PRINT "A MOUSE. "
5090  PRINT : PRINT J$;: INPUT V: PRINT : HOME
5100  IF V < 1 OR V > 4 THEN 5090
5130  ON V GOTO 05140,05170,05190,05210
5140  PRINT "DON'T WORRY ABOUT A MOUSE--A RABBIT IS ": PRINT "MUCH LARGER."
5160 E = E - 10: RETURN
5170  PRINT "GOOD IDEA.  A MOUSE COULDN'T HURT YOU."
5180  RETURN
5190  PRINT "YOU WOULD NOT EAT A MOUSE."
5200  RETURN
5210  PRINT "THE MOUSE SCURRIED INTO ITS HOLE BEFORE"
5220  PRINT "YOU GOT CLOSE."
5230  RETURN
5240  PRINT "NOW YOU SEE A ROOT."
5245  PRINT : PRINT J$;: INPUT V: PRINT : HOME
5250  IF V < 1 OR V > 4 THEN 5245
5280  ON V GOTO 05380,05320,05350,05300
5290  PRINT
5300  PRINT "WHY CHASE A ROOT---BE SENSIBLE."
5310  GOTO 05160
5320  PRINT "THAT'S UP TO YOU.  BUT BEING A MOUSE,":
      PRINT "YOU SHOULD REALLY ENJOY EATING ROOTS."
5340  RETURN
5350  PRINT "GOOD, THIS IS ONE OF YOUR FAVORITES. ":
      PRINT "HOPE YOU ARE ENJOYING IT."
5370 E = E + 15:I = I - 10: RETURN
5380  PRINT "WHY RUN FROM A ROOT? ...THINK..."
5390 E = E - 15: RETURN
5400  PRINT "NOW YOU MEET A GOPHER."
5401  PRINT : PRINT J$;: INPUT V: PRINT : HOME
5410  IF V < 1 OR V > 4 THEN 5401
5440  ON V GOTO 05460,05480,05500,05550
5450  PRINT
5460  PRINT "YOU DON'T HAVE TO RUN FROM IT."
5470  GOTO 05490
5480  PRINT "YOU ARE IGNORING THE GOPHER AND IT": PRINT "RAN AWAY."
5490 E = E - 10: RETURN
5500 R =  INT ( RND (1) * 100) + 1 + I
5510  IF R > 90 THEN 05800
5520  PRINT "YOU GOT IT EVEN THOUGH IT DID NOT PUT":
      PRINT "UP A GOOD FIGHT, AND SLIGHTLY": PRINT "INJURED YOU."
5540 I = I - 5:E = E + 15: RETURN
5550  PRINT "THAT WAS NOT A GOOD CHOICE."
5560  GOTO 05490
5570  PRINT "YOU TURN YOUR HEAD AROUND AND  NOTICE"
5580  PRINT "A FOX IS BEHIND YOU."
5590  PRINT : PRINT J$;: INPUT V: PRINT : HOME
5600  IF V < 1 OR V > 4 THEN 5590
```

```
5630  ON V GOTO 05650,05680,05700,05760
5650  PRINT "YOU'RE MUCH LARGER THAN A FOX."
5660  PRINT "IT WON'T HURT YOU."
5670 E = E - 15: RETURN
5680  PRINT "OKAY."
5690  RETURN
5700 R =  INT ( RND (1) * 100) + 1 + I
5710  IF R >  = 70 THEN 05800
5720  PRINT "YOU GOT IT."
5730 E = E + 15
5740 I = I - 10
5750  RETURN
5760  PRINT "YOU CHASED IT BACK INTO ITS DEN AND YOU": PRINT "CAN'T CHASE IT ANY FARTHER."
5780 E = E - 15
5790  RETURN
5800  PRINT "WELL, YOU DIDN'T GET IT, BUT KEEP YOUR": PRINT "HOPES UP, YOU LL HAVE ANOTHER CHANCE."
5820  RETURN
5830  PRINT : ON A GOTO 5840,5840,6070,6290
5840  PRINT "THERE IS A GRASSY FIELD AHEAD OF YOU.."
5850  PRINT : PRINT J$;: INPUT V: PRINT : HOME
5860  IF V < 1 OR V > 4 THEN 5850
5890  ON V GOTO 5910,5950,5990,6040
5900  PRINT
5910  PRINT "THERE IS NO REASON FOR YOU TO "
5920  PRINT "RUN FROM A FIELD OF GRASS."
5930 E = E - 15
5940  RETURN
5950  PRINT "IGNORING THE FIELD IS FINE , BUT IT HAD": PRINT "FOOD FOR YOU.  YOU MISSED IT."
5970 E = E - 5
5980  RETURN
5990  PRINT "GOOD..A GRASSY FIELD IS GOOD FOR ": PRINT "A ";F$;"."
6000  PRINT "NOW THAT YOU'VE HAD A FULL MEAL,": PRINT "IT IS TIME TO MOVE ON."
6010 E = E + 15
6020 I = I - 15
6030  RETURN
6040  PRINT "DON'T CHASE A FIELD....": PRINT "IT MAKES MOTHER NATURE MAD "
6050 E = E - 10
6060  RETURN
6070  PRINT "THERE IS A BUSH OF BERRIES NEAR."
6080  PRINT : PRINT J$;: INPUT V: PRINT : HOME
6090  IF V < 1 OR V > 4 THEN 6080
6120  ON V GOTO 06140,06180,06240,06210
6140  PRINT "PLEASE BE SENSIBLE..YOU DON T NEED TO ": PRINT "RUN FROM BERRIES."
6160 E = E - 15
6170  RETURN
6180  PRINT "WHY IGNORE THE BERRIES..A FOX LIKES": PRINT "THEM ALOT."
6190 E = E - 5
6200  RETURN
6210  PRINT "A FOX WOULD NOT CHASE BERRIES."
6220 E = E - 15
6230  RETURN
6240  PRINT "VERY GOOD...YOU LIKE BERRIES.": PRINT "LET S MOVE ON..."
6260 E = E + 15
6270 I = I - 15
6280  RETURN
6290  PRINT "NOW THERE IS A GRASSY FIELD NEAR YOU."
6300  PRINT : PRINT J$;: INPUT V: PRINT : HOME
6310  IF V < 1 OR V > 4 THEN 6300
6340  ON V GOTO 6360,6400,6420,6460
6350  PRINT
6360  PRINT "I DON'T LIKE THE COLOR GREEN EITHER..": PRINT "BUT TO RUN FROM GREEN GRASS IS NOT": PRINT "A GOOD MOVE."
6380 E = E - 15
6390  RETURN
6400  PRINT "GOOD...LET'S MOVE ON."
6410  RETURN
6420  PRINT "EVEN THOUGH GRASS IS NOT ON YOUR DIET,"
6430  PRINT "IT DID SETTLE YOUR STOMACH AND IT MADE": PRINT "YOU FEEL BETTER."
6440 E = E + 10
6450  RETURN
6460  PRINT "ARE YOU TRYING TO CHASE GRASS???": PRINT "IT TAKES A LOT OF ENERGY."
6480 E = E - 15
6490  RETURN
6500  PRINT : ON A GOTO 6510,6690,6990,7290
6510  PRINT "ON YOUR TRAVELS THROUGH THE WOODS,": PRINT "YOU COME ACROSS A RACCOON."
6520  PRINT : PRINT J$;: INPUT V: PRINT : HOME
6530  IF V < 1 OR V > 4 THEN 6520
6560  ON V GOTO 6570,6610,6630,6630
6570  PRINT "THAT WAS SMART, BECAUSE RACCOONS ARE ": PRINT "MORE FEROCIOUS THAN MOST PEOPLE THINK."
6590 E = E - 15
6600  RETURN
6610 R =  INT ( RND (1) * 100) + 1 + I
6620  IF R > 20 THEN 06670
6630  PRINT "YOU WEREN'T LUCKY THIS TIME---": PRINT "THE RACCOON ATTACKED AND YOU WERE": PRINT "KILLED FIGHTING HIM."
6650 B$ = "RACCOON"
6660  GOTO 08010
6670  PRINT "DON T TAKE A CHANCE LIKE THAT...": PRINT "YOU COULD GET KILLED."
6680  RETURN
6690  PRINT "THERE IS A HAWK FLYING ABOVE, AND IT ": PRINT "HAS AN EYE ON YOU."
```

```
6700  PRINT : PRINT J$;: INPUT V: PRINT : HOME
6710  IF V < 1 OR V > 4 THEN 6700
6740  ON V GOTO 6750,6900,6960,6940
6750 R =  INT ( RND (1) * 100) + 1 + I
6760  IF R > 70 THEN 06800
6770  PRINT "YOU WERE LUCKY TO HAVE ESCAPED INTO ": PRINT "THE WOODS."
6780 E = E - 15
6790  RETURN
6800  IF R > 95 THEN 06860
6810  PRINT "YOU WERE JUST CAUGHT BY ONE OF ITS CLAWS": PRINT "AND YOU WERE SERIOUSLY HURT."
6830 E = E - 15
6840 I = I + 15
6850  RETURN
6860  PRINT "YOU WEREN'T LUCKY THIS TIME,  ": PRINT "THE HAWK SWOOPED DOWN AND CAUGHT YOU."
6880 B$ = "HAWK"
6890  GOTO 08010
6900 R =  INT ( RND (1) * 100) + 1 - I
6910  IF R >  = 75 THEN 06930
6920  GOTO 06860
6930  PRINT "YOU WERE VERY LUCKY, YOU DIDN'T HAVE ": PRINT "MUCH OF A CHANCE AGAINST HIM, BUT"
6940  PRINT "IT SEEMS AS THROUGH YOU CAME THROUGH."
6950  RETURN
6960  PRINT "THERE IS NO WAY YOU COULD HAVE GOTTEN ": PRINT "TO IT.  REMEMBER YOU ARE A MOUSE. THE"
6970  PRINT "HAWK SWOOPED DOWN AND GOT YOU."
6980  GOTO 06880
6990  PRINT "THERE IS A BIG BEAR IN FRONT OF YOU."
7000  PRINT : PRINT J$;: INPUT V: PRINT : HOME
7010  IF V < 1 OR V > 4 THEN 7010
7040  ON V GOTO 07050,07190,07250,07250
7050 R =  INT ( RND (1) * 100) + 1 + I
7060  IF R <  = 90 THEN 07150
7070  IF R <  = 95 THEN 07110
7080  PRINT "YOU WERE SLOW AND THE BEAR GOT YOU."
7090 B$ = "BEAR"
7100  GOTO 08010
7110  PRINT "YOU WERE INJURED BY THE BEAR'S CLAW."
7120 I = I + 15
7130 E = E - 15
7140  RETURN
7150  PRINT "YOU WERE FAST ENOUGH TO GET AWAY IN ": PRINT "TIME BEFORE HE CAUGHT YOU. "
7170 E = E - 15
7180  RETURN
7190 R =  INT ( RND (1) * 100) + 1 + I
7200  IF R >  = 97 THEN 07220
7205  PRINT "YOU WERE KILLED BY IGNORING THE BEAR."
7210  GOTO 07090
7220  PRINT "YOU WERE VERY LUCKY NOT TO HAVE BEEN ": PRINT "CAUGHT BUT THE BEAR WAS SCARED OFF."
7240  RETURN
7250  PRINT "THAT WAS BRAVE OF YOU, BUT IT DIDN'T ": PRINT "WORK.  THE BEAR GOT YOU WITH A SWING"
7260  PRINT "OF HIS POWERFUL ARM AND CLAWS. "
7280  GOTO 07090
7290  PRINT "THERE IS A LARGE HAWK IN FRONT OF YOU, "
7300  PRINT : PRINT J$;: INPUT V: PRINT : HOME
7310  IF V < 1 OR V > 4 THEN 7300
7340  ON V GOTO 07350,07390,07410,07410
7350  PRINT "YOU SHOULD HAVE IGNORED IT. YOU ARE A": PRINT "BIG WOLF."
7370 E = E - 20
7380  RETURN
7390  PRINT "GOOD, LET'S MOVE ON..."
7400  RETURN
7410  PRINT "AS YOU RAN TOWARDS THE BIRDS, IT ": PRINT "ESCAPED INTO THE AIR."
7430 E = E - 20
7440  RETURN
7680  IF E <  = 0 THEN 07710
7690  IF I >  = 30 THEN 07750
7700  RETURN
7710  PRINT "   YOU JUST RAN OUT OF ENERGY AND WITHOUT"
7720  PRINT "ANY FOOD YOU STARVED TO DEATH."
7730  PRINT "   VISIT ODELL WOODS AGAIN SOON SOMETIME."
7740  GOTO 07830
7750  PRINT "YOUR INJURIES ARE TOO GREAT AND"
7760 R =  INT ( RND (1) * 2) + 1
7770  IF R = 1 THEN 07810
7780  PRINT "YOU FELL PREY TO A FLOCK OF SCAVENGER"
7790  PRINT "BIRDS."
7800  GOTO 07820
7810  PRINT "YOU BLED TO DEATH IN THE WOODS."
7820  PRINT "COME BACK TO ODELL WOODS AGAIN SOMETIME SOON."
7830  GOTO 07950
7950  PRINT
7960  PRINT "WOULD YOU LIKE TO TRY ANOTHER TIME";: INPUT K$
7970  IF K$ = "N" THEN  END
7980  GOTO 40
8010  VTAB 22
8011  PRINT "PRESS 'RETURN' TO CONTINUE": GET V$
8012  POKE 34,0: HOME : PRINT "ALTHOUGH THE ";B$;" GOT YOU ": PRINT
8020  PRINT "COME BACK TO ODELL WOODS AND TRY AGAIN."
8040  GOTO 07950
```

Swarms

Rand K. Miller



Language: BASIC (DEC 10)

Description: SWARMS is a computer simulation (fancy word for game) that was conceived from the book *The Swarm* by Arthur Herzog. The program puts you in charge of the defense of the entire United States when swarms of ferocious South American hybrid bees suddenly start appearing in different sections of the country. The program is provided with in depth instructions which explain the situation very thoroughly.

My goals for this game were to create a program that was 1) my own creation, not an improvement in somebody else's game, 2) as realistic as possible, 3) not another Star Trek game. Since I wanted this project to be a test of my programming skills, I wrote the instructions first and made the game to follow the instruction as closely as possible. The instructions were updated at the end but the program ended up pretty much what I had planned.

I would like to give special credit to the Albuquerque Public School System who own and operate a DEC-10 computer system just for high school students. Their set up is slightly ahead of the times.

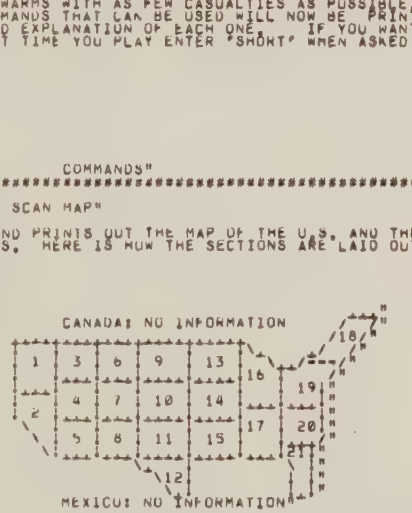
I would also like to give no credit to the Honolulu Public School System which owns and operates nothing for high school students. Their entire city is years behind the times.

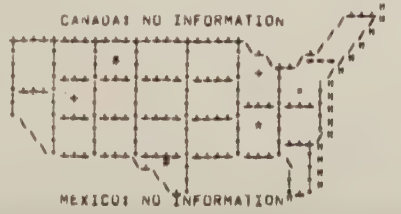
NOTE: When adapting this program for other computer systems, special attention should be paid to the "tab" and "print using" statements (5100-5250) and the margin statement (90) which is not necessarily needed.

Rand K. Miller, 1054 S. King St., Honolulu, HI 96814.

```
DO YOU NEED INSTRUCTIONS ?NO
TIME:    1
ENTER YOUR LAST NAME FOR IDENTIFICATION CHECK.
 ?MILLER
ENTER YOUR CODE WORD FOR LATER VARIFICATION.
 ?SWARMS

COMMAND ?1
TIME:    2
1)       ATTACK SCAN MAP

                                                                    ___
                     CANADA: NO INFORMATION                        /  /
                  ___   ___   ___   ___   ___   ___              _/  /
                 1   1 1   1 1   1 1   1 1   1 1   1\_          _/  /
                 1   1 1   1 1 * 1 1   1 1   1   \  \_/---\    /
                 1   1___1___1___1___1___1   1     \_/     \  /
                 1___1   1   1   1   1   1   1   1         \/
                 1   1   1   1   1   1   1___1___1
                 1   1___1___1___1___1___1   1   1
                  \  1   1   1   1   1   1   1   1
                   \ 1   1   1   1   1   1   1___/
                    \1___1___1___1___1___1___1 *1
                              \_  1           \ 1
                                \ 1           1 1
                                 \_1          1 1
                       MEXICO: NO INFORMATION

COMMAND ?2
TIME:    3
2)       ETA REPORT

SECTION ?9
*****************************************
THE BEES WILL ARRIVE AT THE MAJOR CITY IN
        SECTION 9 AT 9 HOURS.
*****************************************

COMMAND ?2
TIME:    4
2)       ETA REPORT

SECTION ?21
*****************************************
THE BEES WILL ARRIVE AT THE MAJOR CITY IN
        SECTION 21 AT 10 HOURS.
*****************************************
*THE SWARM IN SECTION 9 HAS SPREAD TO SECTION 8

COMMAND ?3
TIME:    5

3)       BATTLE PHASE OPTIONS
SECTION ?8
PHASE ?3
PROJECT BRUSH FIRE: PHASE THREE, NOW BEING ATTEMPTED.
*THE PHASE ON SECTION 8 WAS SUCCESSFUL
*THE SWARM IN SECTION 8 IS READY TO BE DESTROYED
*THE SWARM IN SECTION 9 HAS SPREAD TO SECTION 10
*THE SWARM IN SECTION 21 HAS SPREAD TO SECTION 20

COMMAND ?3
TIME:    6

3)       BATTLE PHASE OPTIONS
SECTION ?20
PHASE ?3
PROJECT BRUSH FIRE: PHASE THREE, NOW BEING ATTEMPTED.

COMMAND ?3
TIME:    7

3)       BATTLE PHASE OPTIONS
SECTION ?10
PHASE ?3
PROJECT BRUSH FIRE: PHASE THREE, NOW BEING ATTEMPTED.
*THE PHASE ON SECTION 10 WAS SUCCESSFUL

COMMAND ?3
TIME:    8
```

```
00010 PRINT"DO YOU NEED INSTRUCTIONS";
00020 INPUT I$
00030 PRINT"TAME:    1"
00040 PRINT"ENTER YOUR LAST NAME FOR IDENTIFICATION CHECK,"
00050 INPUT N$
00060 PRINT"ENTER YOUR CODE WORD FOR LATER VARIFICATION,"
00070 INPUT C$
00080 IF I$<>"YES" THEN 3570
00090 MARGIN 100
00100 PRINT
00110 PRINT
00120 PRINT
00130 PRINT"          IN 1956 A APICULTURIST  WARWIC  KERR BROUGHT TWENTY AFRICAN"
00140 PRINT"QUEEN BEES INTO BRAZIL.  THOUGH IT IS KNOWN TO BE FEROCIOUS, THE"
00150 PRINT"AFRICAN BEE PRODUCES 30  PERCENT MORE MONEY THAN EITHER THE ITALIAN"
00160 PRINT"OR GERMAN BEE THAT LONG DOMINATED BRAZILIAN BEEKEEPING, KERR PLAN-"
00170 PRINT"NED TO CROSS-BREED HIS AFRICANS TO PRODUCE A MORE GENTLE BEE.  WHAT"
00180 PRINT"HE OBTAINED, HOWEVER, WAS A BEE WITH A DISPOSITION SO NASTY THAT IT"
00190 PRINT"THREATENED THE LIVES  AND LIVELIHOOD  OF ALMOST EVERY  BEEKEEPER IN"
00200 PRINT"E STATES OF BRAZIL.  IN JUNE 1968, THE AGRARIAN DEFENSE DERECTORATE"
00210 PRINT"IN LIMA,  PERU, BEGAN EMERGENCY  MEASURES TO PREVENT AN INVASION OF"
00220 PRINT"THE DEADLY AFRILAN BEES FROM BRAZIL.  THE BEES QUICKLY SPREAD OVER"
00230 PRINT"AN AREA ABOUT  EQUAL TO THE SIZE OF THE CONTINENTAL  UNITED STATES."
00240 PRINT"SWARMS OF THESE FEROCIOUS AFRICAN HONEY BEES ARE NOW  MOVING TOWARD"
00250 PRINT"THE UNITED STATES AT THE RATE OF ABOUT 200 MILES A YEAR,  AND THERE"
00260 PRINT"SEEMS TO BE NO NATURAL  BARRIER TO BLOCK THEM, ACCORDING TO A STUDY"
00270 PRINT"FINANCED BY THE DEPARTMENT OF AGRICULTURE.  THIS GAME IS A SIMULA-"
00280 PRINT"TION OF  WHAT MIGHT HAPPEN  WHEN AND IF THE  BEES REACH THE  UNITED"
00290 PRINT"STATES.  IN THIS GAME ONE PERSON IS IN CHARGE OF THE DEFENSE OF THE"
00300 PRINT"ENTIRE UNITED STATES.  THAT ONE PERSON, THROUGH THE USE OF A COMPU-"
00310 PRINT"TER, HAS TO KEEP TRACK OF THE BEES AND TRY TO COMBAT THEM.  COMPU-"
00320 PRINT"TER RADAR  SUBSTATIONS ARE  POSITIONED IN 21 STRATEGIC AREAS OF THE"
00330 PRINT"UNITED  STATES TO SHOW ANY  SWARMS IN THOSE AREAS.  THE SWARMS CAN"
00340 PRINT"AND  WILL GROW AND SPREAD IF THEY ARE NOT COMBATTED.  THEY ARE ALSO"
00350 PRINT"CONSTANTLY  MOVING TOWARD MAJOR CITIES.  EACH SECTION HAS ONE CITY"
00360 PRINT"THAT IS CONSIDERED  MAJOR.  THE SIZE OF THE  SWARMS ARE SHOWN ON AN"
00370 PRINT"MAP AND THE ESTIMATED TIME OF ARRIVAL (ETA) OF THE BEES TO A  MAJOR"
00380 PRINT"CITY IS  PRINTED OUT IN CHART FORM.  THE OBJECT OF THIS GAME IS TO"
00390 PRINT"DESTROY ALL OF THE SWARMS WITH AS FEW CASUALTIES AS POSSIBLE.  ALL"
00400 PRINT"OF THE COMPUTER  COMMANDS THAT CAN BE USED WILL NOW BE  PRINTED OUT"
00410 PRINT"ALONG WITH A DETAILED EXPLANATION OF EACH ONE.  IF YOU WANT SHORT"
00420 PRINT"INSTRUCTIONS THE NEXT TIME YOU PLAY ENTER 'SHORT' WHEN ASKED IF YOU"
00430 PRINT"WANT INSTRUCTIONS."
00440 PRINT
00450 PRINT
00460 PRINT
00470 PRINT
00480 PRINT
00490 PRINT
00500 PRINT
00510 PRINT
00520 PRINT"                              COMMANDS"
00530 PRINT"###############################################################"
00540 PRINT
00550 PRINT"COMMAND 1:    ATTACK SCAN MAP"
00560 PRINT
00570 PRINT"          THIS COMMAND PRINTS OUT THE MAP OF THE U.S. AND THE"
00580 PRINT"          BEE ATTACKS.  HERE IS HOW THE SECTIONS ARE LAID OUT."
00590 PRINT
00600 PRINT
00610 PRINT
00620 PRINT
00630 PRINT
00640 PRINT
00650 PRINT
00660 PRINT
00670 PRINT
00680 PRINT
00690 PRINT
00700 PRINT
00710 PRINT
00720 PRINT
00730 PRINT
00740 PRINT
00750 PRINT
00760 PRINT
00770 PRINT
00780 PRINT
00790 PRINT
00800 PRINT
00810 PRINT
00820 PRINT"          THIS IS WHAT YOU MIGHT SEE ON AN ATTACK SCAN MAP."
00830 PRINT
00840 PRINT
00850 PRINT
00860 PRINT
00870 PRINT
00880 PRINT
00890 PRINT
00900 PRINT
00910 PRINT
00920 PRINT
00930 PRINT
00940 PRINT
00950 PRINT
00960 PRINT
00970 PRINT
00980 PRINT
00990 PRINT
```





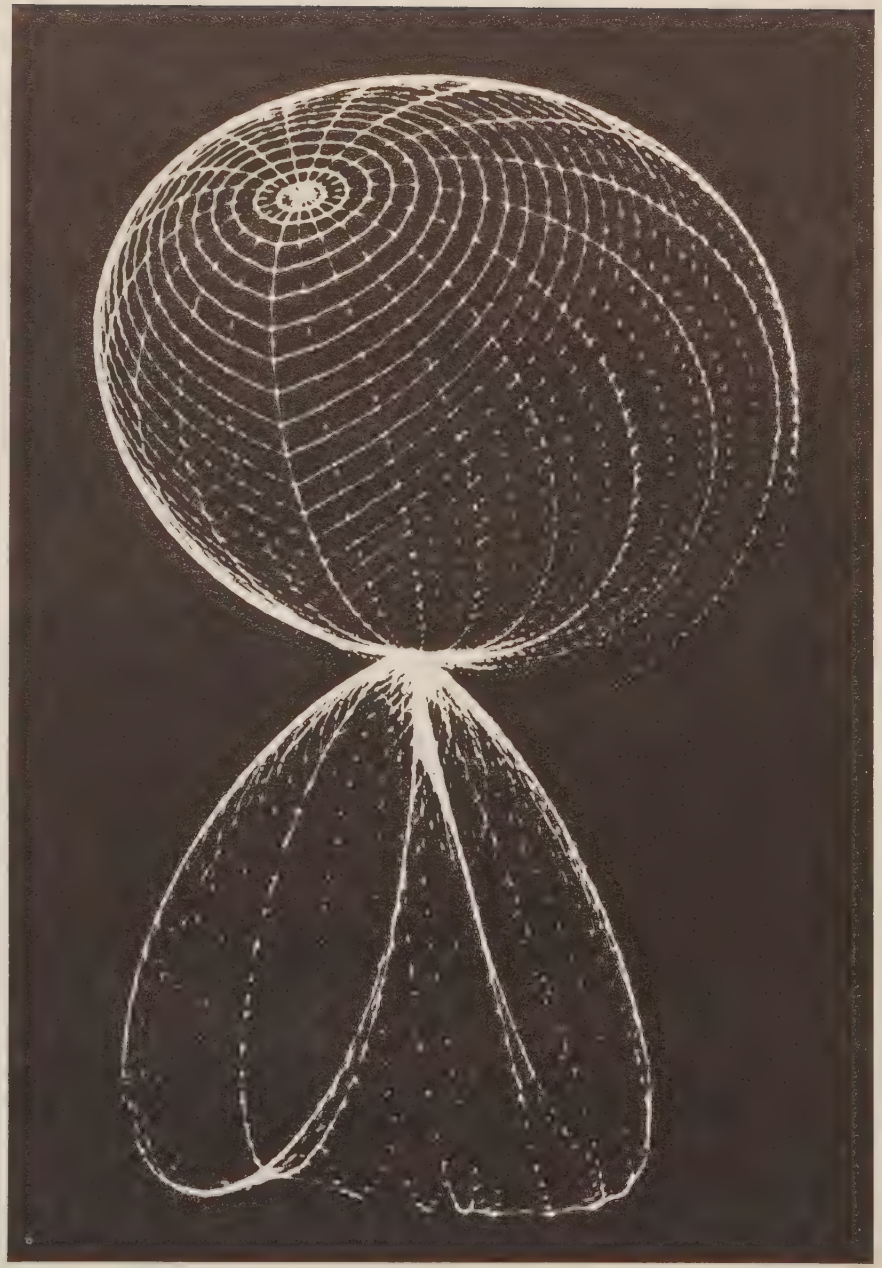

```
01000 PRINT
01010 PRINT
01020 PRINT
01030 PRINT
01040 PRINT"                *      SWARM LARGER THAN 50,000"
01050 PRINT"                #      SWARM BETWEEN 10,000 AND 50,000"
01060 PRINT"                +      SWARM BETWEEN 1,000 AND 10,000"
01070 PRINT"                .      SWARM LESS THAN 1,000 (IS REDUCED ENOUGHT TO BE"
01080 PRINT"                       TOTALLY DESTROYED)"
01090 PRINT
01100 PRINT"###############################################################"
01110 PRINT
01120 PRINT"COMMAND 2:         ETA REPORT"
01130 PRINT
01140 PRINT"          THIS COMMAND ENABLES YOU TO FIND OUT WHEN THE BEES ARE"
01150 PRINT"          ESTIMATED TO REACH A MAJOR CITY.  TO USE THIS COMMAND"
01160 PRINT"          FOR A PARTICULAR SECTION, ENTER 'THIS COMMAND NUMBER(2)'"
01170 PRINT"          WHEN THE COMPUTER PRINTS 'COMMAND?'.  THE COMPUTER WILL"
01180 PRINT"          THEN PRINT OUT 'SECTION?'.  ENTER THE SECTION NUMBER"
01190 PRINT"          THAT YOU WANT.  IF YOU WANT TO FIND THE ETA FOR ALL "
01200 PRINT"          MAJOR CITIES ENTER ZERO(0) AFTER THE COMPUTER PRINTS"
01210 PRINT"          'SECTION?'.  AFTER THE BEES ENTER A CITY YOU WILL"
01220 PRINT"          HAVE 6 COMPUTER HOURS TO COMBAT THEM BEFORE THEY DES-"
01230 PRINT"          STROY ALL OF THE INHABITANTS.  YOU CAN EVACUATE THE "
01240 PRINT"          CITY, HOWEVER, WHICH IS EXPLAINED AS A LATER COMMAND."
01250 PRINT"          EVERY TIME YOU EXECUTE A COMMAND, ONE COMPUTER HOUR IS"
01260 PRINT"          COUNTED.  EACH SECTION HAS ONLY ONE CITY THAT CONSID-"
01270 PRINT"          ERED MAJOR, AND THAT IS THE ONLY CITY YOU SHOULD BE "
01280 PRINT"          CONCERED WITH.  ONCE THE BEES ENTER A CITY, URBAN DEF-"
01290 PRINT"          ENSES MUST BE USED, WHICH ARE EXPLAINED LATER."
01300 PRINT
01310 PRINT"###############################################################"
01320 PRINT
01330 PRINT"COMMAND 3:         BATTLE PHASE OPTIONS"
01340 PRINT
01350 PRINT"          THIS COMMAND IS WHAT YOU USE TO COMBAT THE BEES.  EACH"
01360 PRINT"          OPTION IS EXPLAINED FULLY BELOW.  HERE ARE WHAT YOUR "
01370 PRINT"          OPTIONS ARE."
01380 PRINT"                         BATTLE PHASE OPTIONS"
01390 PRINT
01400 PRINT"                    PHASE ONE:      BEE COCKTAIL"
01410 PRINT"                    PHASE TWO:      PROJECT QUEEN"
01420 PRINT"                    PHASE THREE:    PROJECT BRUSH FIRE"
01430 PRINT"                    PHASE FOUR:     PROJECT STERILE MALE"
01440 PRINT"                    PHASE FIVE:     DESTRUCTION"
01450 PRINT"                    PHASE SIX:      URBAN DEFENSES"
01460 PRINT
01470 PRINT"          EACH OF THESE COMMAND COMBATS THE BEES IN A DIFFERANT"
01480 PRINT"          WAY.  ONE COMMAND WILL WORK BETTER ON LARGER SWARMS,"
01490 PRINT"          ANOTHER COMMAND WILL WORK BETTER ON SMALLER SWARMS."
01500 PRINT"          ONE COMMAND WORKS WITHIN 1 COMPUTER HOUR.  ANOTHER,"
01510 PRINT"          MAY TAKE 5 HOURS.  THERE IS ALWAYS A CHANCE THAT THE"
01520 PRINT"          PHASE WILL NOT WORK AT ALL.  THE ONLY WAY TO FIND OUT "
01530 PRINT"          WHICH COMMANDS WORK ON WHICH SITUATION IS TO USE THEM."
01540 PRINT"          THE APPROXIMATE TIME IT SHOULD TAKE FOR EACH PHASE TO"
01550 PRINT"          TAKE EFFECT IS PRINTED WITH THE COMMAND.  THE TIME"
01560 PRINT"          COULD VARY BY UP TO 2 HOURS.  AS ONE SWARM IS BEING"
01570 PRINT"          COMBATED ANOTHER SWARM CAN BE BUILDING UP.  THE URBAN"
01580 PRINT"          DEFENSES ONLY WORK IN THE CITY AND SOME COULD HARM OR"
01590 PRINT"          KILL SOME OF THE POPULATION ALONG WITH THE BEES.  "
01600 PRINT"          AFTER YOU ENTER THIS COMMAND(3), THE COMPUTER WILL"
01610 PRINT"          ASK YOU FOR THE REST OF THE INFORMATION IT NEEDS."
01620 PRINT
01630 PRINT
01640 PRINT"                         EXPLANATIONS"
01650 PRINT"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
01660 PRINT
01670 PRINT
01680 PRINT"*PHASE ONE*"
01690 PRINT"          BEE COCKTAIL-                              2 HOURS"
01700 PRINT"----------------------------------------------------------------"
01710 PRINT"          AMERICAN FOULBROOD"
01720 PRINT"          INSECT GROWTH REGULATOR(IGR)"
01730 PRINT"          ANTIPHENOMONE SUBSTANCE(LSD)"
01740 PRINT
01750 PRINT"     THOUSANDS OF PACKETS CONTAINING THE ABOVE SUBSTANCES IN A"
01760 PRINT"HONEY SOLUTION WILL BE DROPPED OVER INFESTED AREAS.  THESE SHOULD"
01770 PRINT"KILL SOME OF THE LARVAE, INTERFERE WITH THE DEVELOPMENT OF OTHERS,"
01780 PRINT"AND UPSET THE BEES' COMMUNILATIONS SYSTEM."
01790 PRINT
01800 PRINT"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
01810 PRINT
01820 PRINT
01830 PRINT"*PHASE TWO*"
01840 PRINT"          PROJECT QUEEN-                             5 HOURS"
01850 PRINT"----------------------------------------------------------------"
01860 PRINT"          LINE-BREARING QUEENS"
01870 PRINT"          NEGATIVE SURVIVAL CHARACTERISTICS"
01880 PRINT
01890 PRINT"     ARTIFICIALLY BRED QUEEN BEES WILL BE DROPPED BY HELICOPTERS"
01900 PRINT"OVER INFESTED AREAS.  THEY SHOULD MATE WITH THE AFRICANS, ENTER THE"
01910 PRINT"HIVES, AND DO BATTLE WITH THE AFRICAN QUEENS.  SOME MUST PREVAIL"
01920 PRINT"IN ORDER TO PASS ON DEFECTS TO THEIR PROGENY, AND IN TURN, THEIR"
01930 PRINT"QUEENS AND DRONES WILL SIMILARLY BRED AND INCREASE THE EFFECTIVE "
01940 PRINT"POPULATION UNTIL ENOUGH OF THE AFRICANS HAVE ENOUGH NEGATIVE SURV-"
01950 PRINT"IVAL CHARACTERISTICS TO MAKE THEM SELF-DESTRUCTIVE."
01960 PRINT
01970 PRINT"+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
01980 PRINT
```

```
PRINT"*PHASE THREE!"
PRINT"          PROJECT BRUSH FIRE-                          1 HOUR"
PRINT"-------------------------------------------------------------------------"
PRINT"          LINE-BEARING HORNERS"
PRINT"          STIMULATE AGGRESSION"
PRINT
PRINT"     HUNDREDS OF MILLIONS OF ARTIFICIALLY-MEANED HORNERS (KEYED"
PRINT"WITH AN ATTACK PHEROMONE) ARE DROPPED ON INFESTED AREAS.  THEY ARE"
PRINT"CHEMICALLY COMPELLED TO SEARCH FOR AND ATTACK ANY AFRICAN BEEHIVE"
PRINT"IN THE VICINITY CAUSING REDUCTION OF THE AFRICAN POPULATION."
PRINT
PRINT"*************************************************************************"
PRINT
PRINT"*PHASE FOUR!"
PRINT"          PROJECT STERILE MALE-                        4 HOURS"
PRINT"-------------------------------------------------------------------------"
PRINT"          LINE-BEARING DRONES"
PRINT"          STERILIZATION"
PRINT
PRINT"     HUNDREDS OF MILLIONS OF ARTIFICIALLY-MEANED DRONES WOULD BE"
PRINT"STERILIZED AND DROPPED FROM THE AIR ON INFESTED AREAS.  MATING WITH"
PRINT"THE AFRICAN QUEENS, THEY WOULD INSERT USELESS SPERM TO FILL UP THE"
PRINT"QUEEN.  THE UNFERTILIZED QUEEN WOULD THEN ONLY PRODUCE DRONE EGGS."
PRINT
PRINT"*************************************************************************"
PRINT
PRINT"*PHASE FIVE!"
PRINT"          DESTRUCTION-                                 3 HOURS"
PRINT"-------------------------------------------------------------------------"
PRINT"          BEE WARDENS"
PRINT
PRINT"     THIS DEFENSE IS TO BE USED ONLY AFTER THE BEE POPULATION"
PRINT"HAS BEEN REDUCED BY ONE OF THE LOWER NUMBER PHASES.  TRAINED BEE"
PRINT"WARDENS WILL ENTER THE FOREST IN A LIMITED AREA AND FINISH OFF"
PRINT"THE BEES THAT ARE LEFT."
PRINT"*************************************************************************"
PRINT
PRINT"*PHASE SIX!"
PRINT"          URBAN DEFENSES-                              1 HOUR EACH"
PRINT"-------------------------------------------------------------------------"
PRINT"          ISOLATION DEFENSES"
PRINT"          FINAL RESORT"
PRINT
PRINT"     A) FLIGHT PATTERNS"
PRINT"          COATED METAL FOIL STRIPS ARE DROPPED ON SWARM TO SET"
PRINT"          UP VIBRATIONS IN THE AIR TO DISTURB BEES FLIGHT PAT-"
PRINT"          TERNS."
PRINT"     B) SONIC BOOM"
PRINT"          F-111A'S FLY THROUGH SWARM OF BEES AT SONIC SPEEDS"
PRINT"          TO COMBINE SONIC BOOM,JET EXHAUST,AND JET STREAM,"
PRINT"     C)SUPER-SONIC BEAMS"
PRINT"          SUPER-SONIC BEAM SYNTHESIZER, ORIGINALLY MEANT TO CONFUSE"
PRINT"          THE GUIDANCE SYSTEMS OF SAM MISSILES, ARE PLACED IN STRA-"
PRINT"          TEGIC AREAS AROUND THE CITY.  THE BEES WON'T BE KILLED"
PRINT"          BUT THEIR SOLAR AND GRAVITATIONAL ORIENTATION SHOULD BE"
PRINT"          DISRUPTED ENOUGH TO CAUSE THEM TO FLY INTO BUILDINGS OR"
PRINT"          DIVE INTO THE GROUND."
PRINT"     D)POLLUTION"
PRINT"          THOUSANDS OF CARS ARE POSITIONED ALONG MAJOR TRAFFIC"
PRINT"          ARTERIES OF THE CITY.  THEY ARE WIRED TO BE STARTED BY"
PRINT"          REMOTE CONTROL.  THE BEES SHOULD DIE FROM THE POLLUTION."
PRINT"     E)METHYL PARATHION"
PRINT"          SMALL CROP DUSTERS ARE SENT UP TO SPRAY A METHYL PARA-"
PRINT"          THION BARRIER IN THE PATH OF THE BEES."
PRINT"     F)FIRE WALL"
PRINT"          KEROSENE FILLED TANKER TRUCKS ARE IGNITED IN THE PATH OF"
PRINT"          THE BEES.  SOME BEES SHOULD BE BURNED AND OTHERS SHOULD"
PRINT"          BE CARRIED UP BY THE HOT AIR TO ALTITUDES BEYOND THEIR"
PRINT"          CAPABILITIES."
PRINT"     G)STROBE LIGHT"
PRINT"          A HIGH-INTENSITY STROBE LIGHT IS POSITIONED ON TOP OF A"
PRINT"          HIGH BUILDING.  IT IS SURROUNDED BY A HIGH-VOLTAGE WIRE"
PRINT"          MESH.  THE BEES SHOULD BE ATTRACTED TO THE STROBE AND"
PRINT"          ELECTROCUTED."
PRINT
PRINT"*************************************************************************"
PRINT
PRINT"COMMAND 4:        EVACUATION"
PRINT
PRINT"          THIS COMMAND IS USED TO EVACUATE A MAJOR CITY IN ANY"
PRINT"          SECTION.  THE EVACUATION PROCESS TAKES ABOUT 5 COMPUTER"
PRINT"          HOURS TO COMPLETE.  IT IS NEVER REQUIRED FOR YOU TO"
PRINT"          EVACUATE A CITY FOR ANY REASON, THE COMPUTER WILL NOT"
PRINT"          ALLOW THE EVACUATION OF A MAJOR CITY IN A SECTION IN"
PRINT"          WHICH NO SWARMS ARE REPORTED.  NO BATTLE PHASE OPTIONS"
PRINT"          CAN BE USED IN A SECTION THAT IS BEING EVACUATED UNTIL"
PRINT"          IT IS COMPLETELY EVACUATED."
PRINT
PRINT"*************************************************************************"
PRINT
PRINT"COMMAND 5:        NUCLEAR DESTRUCTION"
PRINT
PRINT"          ANY CITY CAN BE DESTROYED COMPLETELY BY USING THIS COM-"
PRINT"          MAND.  THE CITY DOES NOT NECESSARILY HAVE TO BE EVAL-"
PRINT"          UATED TO EXECUTE THIS COMMAND.  IF THE BEES HAVE NOT"
PRINT"          ENTERED THE MAJOR CITY THE COMPUTER WILL NOT ALLOW IT"
PRINT"          TO BE DESTROYED USING THIS METHOD.  BEFORE USING THIS"
PRINT"          COMMAND IT IS FIRST NECESSARY FOR YOU TO ENTER YOUR "
PRINT"          NAME AND CODEWORD EXACTLY AS YOU DID AT THE BEGINNING,"
PRINT"          IF THE POPULATION IS EVACUATED ONLY THE BEES ARE DES-"
PRINT"          TROYED AND THE POPULATION WILL RETURN TO THE CITY AFTER"
PRINT"          ABOUT 7 COMPUTER HOURS.  IF THE POPULATION IS NOT EVAC-"
PRINT"          UATED IT IS DESTROYED WITH THE BEES AND THE ENTIRE SEC-"
PRINT"          TION BECOMES UNINHABITED.  THE BEES WILL NOT ENTER IN-"
PRINT"          TO A SECTION THAT IS COMPLETELY UNINHABITED, SO THAT"
PRINT"          SECTION WILL BE ENTIRELY EMPTY FOR THE REMAINDER OF THE"
PRINT"          GAME"
PRINT
PRINT"*************************************************************************"
PRINT
PRINT"COMMAND 6:        CASUALTY REPORT"
PRINT
PRINT"          THIS COMMAND PERMITS YOU TO FIND OUT HOW MANY BEE"
PRINT"          RELATED CASUALTIES THERE ARE IN THE SECTIONS  TO"
PRINT"          USE THIS COMMAND FOR ALL THE SECTIONS ENTER ;ERU(0)"
PRINT"          FOR THE SECTION NUMBER.  A CASUALTY REPORT IS AUTO-"
PRINT"          MATICALLY ISSUED AT THE END OF A GAME UNLESS THE"
PRINT"          GAME CANCEL COMMAND IS USED."
PRINT
PRINT"*************************************************************************"
PRINT
PRINT"COMMAND 7:        COMMAND(SHORT)"
PRINT
PRINT"          THIS COMMAND PRINTS OUT ALL OF THE COMMANDS IN A"
PRINT"          SHORT VERSION WITH NO EXPLANATIONS."
PRINT
PRINT"*************************************************************************"
PRINT
PRINT"COMMAND 8:        CANCEL GAME"
PRINT
PRINT"          THIS COMMAND CANCELS THE GAME YOU ARE PLAYING AND"
PRINT"          THEN ASK YOU IF YOU WANT TO PLAY ANOTHER ONE."
PRINT
PRINT"*************************************************************************"
PRINT
PRINT
PRINT
PRINT
REM     *** CREATE 2 INITIAL SWARMS ***
RANDOMIZE
DIM E(25),S(25),A3(25),U(25),G(25),M(25),D(25),C(25),V(25),K(25),R(25)
A1=INT(RND*21+1)
A2=INT(RND*21+1)
IF A1=A2 GOTO 3410
S(A1)=INT(RND*5+2)
S(A2)=INT(RND*5+2)
E(A1)=INT(RND*9+6)
E(A2)=INT(RND*9+6)
IF I$="SHORT" THEN 8160
REM     *** SUBTRACT ONE HOUR FROM ALL ETA'S ***
I$="NO"
FOR A=1 TO 21
IF E(A)=0 THEN 3550
GOTO 3540
IF U(A)=0 THEN 3750
IF S(A)=1 THEN 3750
IF ABS(U(A))=1 THEN 3660
IF ABS(E(A))=1 THEN 3600
E(A)=E(A)-1
C(A)=C(A)+(17-E(A))
GOTO 3750
IF E(A)=-1 THEN 3640
PRINT"*TIME BEES HAVE ARRIVED IN THE MAJOR CITY IN SECTION"A
E(A)=-1
U(A)=6
U(A)=U(A)-1
GOTO 3750
IF U(A)=-1 THEN 3750
PRINT"*TIME BEES HAVE DESTROYED THE MAJOR CITY IN SECTION"A
U(A)=-1
IF V(A)<>-1 THEN 3720
PRINT"BUT THE POPULATION HAS BEEN EVACUATED"
GOTO 3740
K(A)=1
C(A)=(1000000*(RND*A+1))+C(A)
GOTO 3750
NEXT A
REM     *** ADD ONE UNIT TO TIME ***
T=T+1
REM     *** CREATE NEW SWARM ***
IF T/30<>INT(T/30) THEN 3880
A=INT(RND*21+1)
IF S(A)<>0 THEN 3790
IF K(A)<>0 THEN 3790
IF K(A)<>0 THEN 3790
S(A)=INT(RND*5+2)
E(A)=INT(RND*9+16)
C(A)=INT(RND*10)
PRINT"A NEW SWARM IS REPORTED IN SECTION"A
REM     *** CHECK COMBATED SWARMS ***
FOR A=1 TO 21
IF C<=G(A) THEN 4000
IF S(A)<>1 THEN 4000
S(A)=0
G(A)=0
E(A)=0
M(A)=0
K(A)=0
```

```
03970  U(A)=0
03980  PRINT"*THE SWARM IN SECTION"A"IS TOTALLY DESTROYED."
03990  GOTO 4060
04000  S(A)=S(A)-M(A)
04010  PRINT"*THE PHASE ON SECTION"A"WAS SUCCESSFUL"
04020  IF S(A)>=1 THEN 4040
04030  S(A)=1
04040  G(A)=0
04050  M(A)=0
04060  NEXT A
04070  REM     *** ADD ONE UNIT TO EACH UNCOMBATTED SWARM ***
04080  FOR A=1 TO 21
04090  IF E(A)=1 THEN 4340
04100  IF S(A)=1 THEN 4310
04110  IF S(A)=0 THEN 4340
04120  IF S(A) +1<9 THEN 4280
04130  IF A=1 THEN 4210
04140  IF S(A-1)>0 THEN 4200
04150  IF R(A-1)<>0 THEN 4340
04160  PRINT"*THE SWARM IN SECTION"A"HAS SPREAD TO SECTION"A-1
04170  S(A-1)=3
04180  E(A-1)=INT(RND *10+5)
04190  GO TO 4340
04200  IF A=21 THEN 4340
04210  IF S(A+1)>0 THEN 4340
04220  IF R(A+1)<>0 THEN 4340
04230  PRINT"*THE SWARM IN SECTION"A"HAS SPREAD TO SECTION"A+1
04240  IF R(A+1)<>0 THEN 4340
04250  S(A+1)=3
04260  E(A+1)=INT(RND *10+5)
04270  GOTO 4340
04280  S(A)=S(A)+1
04290  C(A)=C(A)+S(A)
04300  GOTO 4340
04310  IF D(A)=1 THEN 4340
04320  PRINT"*THE SWARM IN SECTION"A"IS READY TO BE DESTROYED"
04330  D(A)=1
04340  NEXT A
04350  REM     *** WINNER CHECK ***
04360  W=0
04370  FOR A=1 TO 21
04380  W=S(A)+W
04390  NEXT A
04400  IF W>=1 THEN 4470
04410  PRINT"     ******"G ALL SWARMS ARE NOW DESTROYED "G******"
04420  PRINT"FINAL STATISTICS ON ATTACKS AND DESTRUCTION ARE NOW "
04430  PRINT"BEING COMPUTED,   "G "G "G "G "G "G  FINAL RESULTS FOLLOW....."
04440  A=0
04450  PRINT
04460  GOTO 7920
04470  REM     *** EVACUATION CHECK ***
04480  FOR A=1 TO 21
04490  IF V(A)<>T THEN 4590
04500  IF U(A)=-1 THEN 4560
04510  IF K(A)>0 THEN 4560
04520  PRINT"*POPULATION IN SECTION"A"IS EVACUATED"
04530  C(A)=C(A)+INT(RND*1F)
04540  V(A)=-1
04550  GOTO 4590
04560  PRINT"*POPULATION IN SECTION"A"WAS DESTROYED BEFORE IT"
04570  PRINT"COULD BE EVACUATED"
04580  V(A)=0
04590  NEXT A
04600  REM     *** RETURN EVACUTATION CHECK ***
04610  FOR A=1 TO 21
04620  IF R(A)<>T THEN 4670
04630  IF R(A)<1 THEN 4670
04640  K(A)=0
04650  R(A)=0
04660  PRINT"*THE POPULATION HAS RETURNED TO THE CITY IN SECTION"A
04670  NEXT A
04680  REM     *** LOSER CHECK ***
04690  Q=0
04700  FOR A=1 TO 21
04710  Q=C(A)+Q
04720  NEXT A
04730  IF Q<75000000 THEN 4830
04740  PRINT"THE BEES HAVE DEVASIATED THE UNITED STATES"G AND"
04750  PRINT"THERE ARE NOW OVER 75 MILLION CASUALTIES.  THE BEES"
04760  PRINT"ARE NOW CONSIDERED TO BE THE VICTORS OVER MODERN TECH-"
04770  PRINT"NOLOGY."G "G "G "G "G "G "G "G "G "G "G "G "G "G "G"
04780  A=0
04790  Q=0
04800  W=0
04810  PRINT
04820  GOTO 7920
04830  RE4     *** COMMAND INPUT ***
04840  PRINT
04850  PRINT"COMMAND";
04860  INPUT C
04870  PRINT"TIME:  "T+1
04880  IF C>0 THEN 4850
04890  ON C GOTO 4910,5400,5610,7040,7260,7880,8150,8420
04900  GOTO 4860
04910  REM     *** MAP PRINT OUT ***
04920  PRINT"1)          ATTACK SCAN MAP"
04930  PRINT
04940  FOR A=1 TO 21
04950  IF S(A)>6 THEN 5070
04960  IF S(A)>4 THEN 5050
04970  IF S(A)>1 THEN 5030
04980  IF S(A)>0 THEN 5010
04990  LET A$(A)=" "
05000  GOTO 5060
05010  LET A$(A)="."
05020  GOTO 5060
05030  LET A$(A)="+"
05040  GOTO 5060
05050  LET A$(A)="*"
05060  GOTO 5060
05070  LET A$(A)="#"
05080  NEXT A
05090  PRINT"-------------------------------------------------------------"
05100  PRINT;TAB(48)"  "
05110  PRINT;TAB(20)"CANADA: NO INFORMATION     /  /"
05120  PRINT USING 5280,A$(18)
05130  PRINT;TAB(15)"|  |  |  |  |  |  |  /"
05140  PRINT USING 5290,A$(1),A$(3),A$(6),A$(9),A$(13)"/"
05150  PRINT USING 5300,A$(16)
05160  PRINT USING 5310,A$(19)
05170  PRINT USING 5320,A$(4),A$(7),A$(10),A$(14)
05180  PRINT USING 5330,A$(2)
05190  PRINT USING 5340,A$(17),A$(20)
05200  PRINT USING 5350,A$(5),A$(8),A$(11),A$(15)
05210  PRINT USING 5360,A$(21)
05220  PRINT;TAB(27)"\   |"TAB(42)"\  |"
05230  PRINT USING 5370,A$(12)
05240  PRINT;TAB(36)"\  |"TAB(42)"|  |"
05250  PRINT;TAB(20)"MEXICO: NO INFORMATION"
05260  PRINT
05270  PRINT"-------------------------------------------------------------"
05280  :                                                  /  /
05290  :                                                 /  /
05300  :                  +--+--+--+--+--+--+--+      \+/--/
05310  :                  |  |  |  |  |  |  |  |       |  /
05320  :                  +--+--+--+--+--+--+--+--+    |  |
05330  :                  |  |  |  |  |  |  |  |  |    |  |
05340  :                  +--+--+--+--+--+--+--+--+    |  |
05350  :                   \ |  |  |  |  |  |  |  |    |  |
05360  :                    \+--+--+--+--+--+--+--+    |  |
05370  :                                              |  |
05380  PRINT
05390  GOTO 3480
05400  REM     *** ETA REPORT ***
05410  PRINT"2)          ETA REPORT"
05420  PRINT
05430  PRINT"SECTION";
05440  INPUT A
05450  PRINT"*****************************************"
05460  IF A<>0 THEN 5490
05470  LET D=-1
05480  FOR A=1 TO 21
05490  IF S(A)=1 THEN 5710
05500  IF R(A)=-1 THEN 5740
05510  IF E(A)=0 THEN 5690
05520  IF U(A)=-1 THEN 5770
05530  IF E(A)=1 THEN 5600
05540  PRINT"THE BEES WILL ARRIVE AT THE MAJOR CITY IN"
05550  PRINT"          SECTION"A"AT"E(A)+T"HOURS."
05560  GOTO 5760
05570  PRINT"THE BEES HAVE ALREADY DESTROYED THE CITY"
05580  PRINT"IN SECTION"A"AND ARE NOW INHABITING IT."
05590  GOTO 5760
05600  PRINT"THE BEES HAVE ARRIVED IN THE CITY IN SEC="
05610  IF V(A)<>-1 THEN 5650
05620  PRINT"TION"A"BUT THE POPULATION HAS BEEN"
05630  PRINT"                  EVACUATED"
05640  GOTO 5760
05650  PRINT"  TION"A"AND THE POPULATION OF THAT CITY"
05660  PRINT"CAN ONLY SURVIVE FOR ABOUT"U(A)"MORE HOURS."
05670  PRINT"    USE OF URBAN DEFENSES IS RECOMENDED."
05680  GOTO 5760
05690  PRINT"     NO SWARMS REPORTED IN SECTION"A
05700  GOTO 5760
05710  PRINT"THE BEES IN SECTION"A"ARE READY TO  BE"
05720  PRINT"              DESTROYED"
05730  GOTO 5760
05740  PRINT"SECTION"A"IS A RADIOACTIVE WASTELAND THAT "
05750  PRINT"     IS COMPLETELY UNPOPULATED"
05760  PRINT"*****************************************"
05770  IF D<>-1 THEN 5790
05780  NEXT A
05790  U=0
05800  GOTO 3480
05810  REM  *** BATTLE PHASE OPTIONS ***
05820  PRINT
05830  PRINT"3)          BATTLE PHASE OPTIONS"
05840  PRINT"SECTION";
05850  INPUT A
05860  PRINT"PHASE";
05870  INPUT P
05880  IF V(A)<1 THEN 5910
05890  PRINT"SECTION"A"IS BEING EVACUATED"
05900  GOTO 3480
05910  IF C(A)>0 THEN 5930
05920  GOTO 5950
05930  PRINT"THE SWARM IN SECTION"A"IS ALREADY BEING COMBATTED"
05940  GOTO 3480
```

```
06940   GOTO 6980
06950   REM *G*
06960   IF RND>.5 THEN 6980
06970   S(A)=S(A)-4
06980   PRINT"THE URBAN DEFENSE IN SECTION"A"WAS SUCESSFUL"
06990   IF S(A)>1 THEN 3480
07000   S(A)=1
07010   D(A)=1
07020   PRINT"*SWARM IN SECTION"A"IS READY TO BE DESTROYED"
07030   GOTO 3480
07040   REM     *** EVACUATION PROCEDURE ***
07050   PRINT
07060   PRINT"4)          EVACUATION PROCEDURE"
07070   PRINT"SECTION";
07080   INPUT A
07090   IF S(A)=0 THEN 7150
07100   IF V(A)>0 THEN 7170
07110   IF V(A)=-1 THEN 7190
07120   IF K(A)>0 THEN 7210
07130   IF U(A)=1 THEN 7210
07140   GOTO 7230
07150   PRINT"COMPUTER FAILSAFE...NO SWARMS REPORTED IN SECTION"A
07160   GOTO 3480
07170   PRINT"SECTION"A"IS BEING EVACUATED ALREADY"
07180   GOTO 3480
07190   PRINT"CITY IN SECTION"A"IS ALREADY EVACUATED"
07200   GOTO 3480
07210   PRINT"PUPULATION IN SECTION"A"HAS BEEN DESTROYED"
07220   GOTO 3480
07230   PRINT"EVACUATION PROCEDURE, NOW IN PROCCESS"
07240   V(A)=S+1
07250   GOTO 3480
07260   REM     *** NUCLEAR DESTRUCTION SEQUENCE ***
07270   PRINT
07280   PRINT"5)          NUCLEAR DESTRUCTION SEQUENCE"
07290   PRINT"PLEASE ENTER YOUR NAME"
07300   INPUT N1$
07310   IF N$<>N1$ THEN 7490
07320   PRINT"PLEASE ENTER YOUR CODEWORD"
07330   INPUT C1$
07340   IF C$<>C1$ THEN 7490
07350   PRINT"POSITIVE IDENTIFICATION CHECK "G"G"G"G"G"G"G"
07360   PRINT"CODEWORD CHECK IS VALID "G"
07370   PRINT"ID SEQUENCE COMPLETED"
07380   PRINT"PLEASE ENTER SECTION "
07390   INPUT A
07400   IF E(A)<>-1 THEN 7440
07410   IF S(A)=0 THEN 7470
07420   PRINT
07430   GOTO 7530
07440   PRINT"COMPUTER FAILSAFE: BEES HAVE NOT ARRIVED IN THE MAJOR "
07450   PRINT"CITY IN SECTION"A
07460   GOTO 3480
07470   PRINT"COMPUTER FAILSAFE: NO SWARM REPORTED IN SECTION"A
07480   GOTO 3480
07490   PRINT"ID SEQUENCE DEFAULT.....START AGAIN"
07500   GOTO 3480
07510   PRINT"NUCLEAR DESTRUCTION ABORTED."
07520   GOTO 3480
07530   PRINT"BOMB IN SECTION"A"IS NOW ACTIVATED"
07540   IF K(A)=1 THEN 7600
07550   IF V(A)=-1 THEN 7600
07560   PRINT"SECTION"A"HAS NOT BEEN EVACUATED"
07570   PRINT"DO YOU WISH TO CONTINUE"
07580   INPUT K$
07590   IF K$<>"YES" THEN 7510
07600   PRINT"PUSH RETURN FOR INSTANTANIOUS DETONATION"
07610   INPUT E$
07620   PRINT"GBOMB "GHAS BEEN "GDETONATED"G "G"
07630   PRINT"SWARM IS DESTROYED"G "G "G"
07640   PRINT"CITY IS DESTROYED "G "G "G"
07650   C(A)=C(A)+2413
07660   S(A)=0
07670   E(A)=0
07680   U(A)=0
07690   D(A)=0
07700   G(A)=0
07710   M(A)=0
07720   IF K(A)=1 THEN 7810
07730   IF V(A)<>-1 THEN 7800
07740   PRINT"THE POPULATION WILL MOVE BACK TO THE"
07750   PRINT"CITY IN SECTION"A"WHEN THE RADIATION"
07760   PRINT"LEVEL HAS DECREASED. "
07770   K(A)=2
07780   R(A)=1+/
07790   GOTO 3480
07800   C(A)=INT(2000000*(RND*A+1))+C(A)
07810   PRINT"*NO SURVIVERS ARE REPORTED IN SECTION"A
07820   PRINT"*THE BEES WILL NOT ENTER A SECTION WITH"
07830   PRINT"NO HUMAN INHABITANTS, SO SECTION"A
07840   PRINT"IS COMPLTELY LACKING LIFE OF ANY KIND"
07850   K(A)=2
07860   R(A)=-1
07870   GOTO 3480
07880   REM     *** CASUALTY REPORT ***
07890   PRINT"  6)          CASUALTY REPORT"
07900   PRINT"SECTION";
07910   INPUT A
07920   PRINT"*****************************************"

05950   IF S(A)=0 THEN 5970
05960   GOTO 5990
05970   PRINT"NO SWARM IS REPORTED IN SECTION"A
05980   GOTO 3480
05990   IF S(A)<>1 THEN 6030
06000   IF P=2 THEN 6070
06010   PRINT"THE DESTRUCTION PHASE SHOULD BE USED IN SECTION"A
06020   GOTO 3480
06030   IF E(A)<>-1 THEN 6070
06040   IF P=2 THEN 6070
06050   PRINT"URBAN DEFENSES SHOULD BE USED IN SECTION"A
06060   GOTO 3480
06070   N=RND
06080   ON P GOTO 6090,6180,6270,6350,6420,6480
06090   REM *PHASE 1*
06100   PRINT"BEE COCKTAIL: PHASE ONE, NOW BEING ATTEMPTED,"
06110   IF N>.95 THEN 3480
06120   G(A)=T+INT(RND*3+1)
06130   IF S(A)>5 THEN 6160
06140   LET M(A)=S(A)=2
06150   GOTO 3480
06160   LET M(A)=S(A)=5
06170   GOTO 3480
06180   REM *PHASE 2*
06190   PRINT"PROJECT QUEEN: PHASE TWO, NOW BEING ATTEMPTED,"
06200   IF N>.92 THEN 3480
06210   G(A)=T+INT(RND *3+4)
06220   IFS(A)>3 THEN 6250
06230   M(A)=1
06240   GOTO 3480
06250   M(A)=S(A)-1
06260   GOTO 3480
06270   REM *PHASE 3*
06280   PRINT"PROJECT BRUSH FIRE: PHASE THREE, NOW BEING ATTEMPTED,"
06290   IF N>.90 THEN 3480
06300   G(A)=T+1
06310   M(A)=2
06320   GOTO 3480
06330   REM *PHASE 4*
06340   PRINT"PROJECT STERILE MALE: PHASE FOUR, NOW BEING ATTEMPTED,"
06350   IF N>.89 THEN 3480
06360   G(A)=T+INT(RND*3+3)
06370   IF S(A)>6 THEN 6400
06380   M(A)=S(A)-1
06390   GOTO 3480
06400   M(A)=4
06410   GOTO 3480
06420   REM *PHASE 5*
06430   IF S(A)<>1 THEN 6010
06440   PRINT"DESTRUCTION: PHASE FIVE, NOW BEING ATTEMPTED,"
06450   IF N>.7 THEN 3480
06460   G(A)=T+INT(RND * 3+2)
06470   GOTO 3480
06480   REM *PHASE 6*
06490   IF E(A)<>-1 THEN 6010
06500   PRINT"URBAN DEFENSE*;
06510   INPUT D$
06520   C(A)=C(A)+INT(200*RND)
06530   IF RND >.8 THEN 3480
06540   IF D$="A" THEN 6630
06550   IF D$="B" THEN 6690
06560   IF D$="L" THEN 6750
06570   IF D$="D" THEN 6810
06580   IF D$="E" THEN 6870
06590   IF D$="F" THEN 6910
06600   IF D$="G" THEN 6950
06610   PRINT"INVALID URBAN DEFENSE: PLEASE START AGAIN."
06620   GOTO 3480
06630   REM *A*
06640   IF S(A)>5 THEN 6670
06650   S(A)=S(A)-1
06660   GOTO 6980
06670   S(A)=2
06680   GOTO 6980
06690   REM *B*
06700   IF S(A)>4 THEN 6730
06710   S(A)=S(A)-3
06720   GOTO 6980
06730   S(A)=S(A)-2
06740   GOTO 6980
06750   REM *C*
06760   IF S(A)>7 THEN 6790
06770   S(A)=S(A)-3
06780   GOTO 6980
06790   S(A)=1
06800   GOTO 6980
06810   REM *D*
06820   IF S(A)>3 THEN 6850
06830   S(A)=1
06840   GOTO 6980
06850   S(A)=S(A)-3
06860   GOTO 6980
06870   REM *E*
06880   IF RND>.5 THEN 6980
06890   S(A)=2
06900   GOTO 6980
06910   REM *F*
06920   IF RND >.4 THEN 6980
06930   S(A)=1
```

```
07930   IF A<>0 THEN 7960
07940   LET U=1
07950   FOR A=1 TO 21
07960   IF C(A)=0 THEN 8060
07970   IF C(A)>=100000 THEN 8020
07980   PRINT"SECTION"A";     "C(A)"CASUALTIES REPORTED"
07990   GOTO 8030
08000   PRINT"SECTION"A";     NO BEE RELATED CASUALTIES"
08010   GOTO 8030
08020   PRINT"SECTION"A";       "C(A)/100000"MILLION CASUALTIES"
08030   PRINT"*****************************************"
08040   F=C(A)+F
08050   IF U<>1 THEN 8120
08060   NEXT A
08070   U=0
08080   IF F<100000 THEN 8110
08090   PRINT"TOTAL CASUALTIES REPORTED: "F/100000"MILLION"
08100   GOTO 8120
08110   PRINT"TOTAL CASUALTIES REPORTED:"F
08120   IF W=0 THEN 8030
08130   F=0
08140   GOTO 3480
08150   REM    *** PRINT COMMANDS IN SHORT ***
08160   PRINT"************************ COMMANDS *************************"
08170   PRINT
08180   PRINT"            1) ATTACK SCAN MAP"
08190   PRINT"            2) EIA REPORT"
08200   PRINT"            3) BATTLE PHASE OPTIONS"
08210   PRINT"                  1. BEE COCKTAIL"
08220   PRINT"                  2. PROJECT QUEEN"
08230   PRINT"                  3. PROJECT BRUSH FIRE"
08240   PRINT"                  4. PROJECT STERILE MALE"
08250   PRINT"                  5. DESTRUCTION"
08260   PRINT"                  6. URBAN DEFENSES"
08270   PRINT"                        A) FLIGHT PATTERNS"
08280   PRINT"                        B) SONIC BOOM"
08290   PRINT"                        C) SUPER-SONIC BEAMS"
08300   PRINT"                        D) POLLUTION"
08310   PRINT"                        E) MYINYL PARATHION"
08320   PRINT"                        F) FIRE WALL"
08330   PRINT"                        G) STROBE LIGHT"
08340   PRINT"            4) EVACUATION PROCEDURES"
08350   PRINT"            5) NUCLEAR DESTRUCTION"
08360   PRINT"            6) CASUALTY REPORT"
08370   PRINT"            7) COMMANDS (SHORT)"
08380   PRINT"            8) CANCEL GAME"
08390   PRINT
08400   GOTO 3480
08410   REM    *** CANCEL GAME ***
08420   PRINT"GAME CANCELLED"
08430   PRINT"DO YOU WANT TO PLAY AGAIN";
08440   INPUT X$
08450   IF X$="NO" THEN 8630
08460   PRINT"ANOTHER GAME IS NOW BEING SET UP...."
08470   MAT E=ZER
08480   MAT S=ZER
08490   MAT U=ZER
08500   MAT G=ZER
08510   MAT M=ZER
08520   MAT O=ZER
08530   MAT C=ZER
08540   MAT V=ZER
08550   MAT K=ZER
08560   MAT R=ZER
08570   A=W=T=P=C=U=F=0
08580   PRINT".........READY!"
08590   PRINT
08600   PRINT
08610   PRINT
08620   GOTO 3370
08630   PRINT
08640   PRINT
08650   PRINT
08660   PRINT"     THIS GAME WAS BASED ON THE BOOK THE SWARM, BY AUTHUR HERZOG."
08670   PRINT"IT WAS CREATED AND PRODUCED BY MAND K. MILLER.  IT WAS ORIGINAL-"
08680   PRINT"LY RELEASED IN ALBUQUERQUE, N.M. ON APRIL 1,1976.     THANK YOU."
08690   END
```



Genetic Calculations in Basic

Peter Koski

Unlike Fortran, Basic allows for logical manipulation of alphanumeric information (strings) as well as complex mathematical calculations. Thus, by combining the two, very powerful problem solving is available to the user. The following program takes advantage of both of these, particularly Basic's ability to handle complex strings.

Most work in genetics involves the cross between two parents, and the subsequent study of the resulting off-spring. When more than two or three genes are involved, or when sex-linked characteristics are analyzed, the results can be staggering. Often, the time spent in reaching these results is also staggering.

Peter Koski, 27 Dogwood Dr., Allendale, NJ 07401.

Using The Program

Referring to the sample run, parent one has been entered as being a heterozygous brown-eyed, homozygous tall male; parent two as a heterozygous brown-eyed, heterozygous tall female. The program then promptly returns the "gamete number" (total different gene combinations for each parent). The gametes of each parent are then printed out along with a list of the resulting off-spring.

The program doesn't end here, however, as the "Pertinent Statistics Subprogram" takes over the job of sorting out this mess. By entering one or two genes of interest, the subprogram will count the number of occurrences of each gene or combination of two genes, as well as compute the gene frequency.

This program was written for use on an Ohio Scientific Challenger II with a printer attached to the serial port. The printer is optional, as a subroutine within the program monitors the "ESC" key to provide alternate start/stop control of the listings.

The program runs fine with 8K, however, more memory would permit crosses of greater complexity. The program as it is presented here allows for a maximum of 128 off-spring (only 16 are possible when running with 8K) in crosses involving 6 genes or less. To increase these limits, the DIMension statements must be changed, as well as the housekeeping For/Next loop in statements 1790-1810.

```
990 REM    >>>>> AUTHORED BY PETER KOSKI <
995 REM    >>>>>       APRIL, 1979
1000 DIM G1$(6), G2$(6), A$(16,6), B$(16,6), OS$(128)
1010 FOR CLR=1 TO 25: PRINT: NEXT CLR
1020 PRINT "            ** GENETICS PROGRAM **"
1030 PRINT
1040 PRINT "    OFFSPRING GENERATION AND CORRESPONDING STATISTICS"
1050 PRINT:PRINT "                    ------":PRINT
1060 PRINT "SHIFT LOCK MUST BE RELEASED.  USE LEFT SHIFT TO ENTER CR";
1070 PRINT "PITALS":PRINT
1075 PRINT "                    --------":PRINT
1077 PRINT "DURING LISTINGS, SCROLL MAY BE STOPPED / STARTED USING ";
1078 PRINT "'ESC' KEY"
1080 PRINT:PRINT "                    --------":PRINT
1090 PRINT:INPUT "ENTER PARENT 1";P1$
1100 PRINT:INPUT "ENTER PARENT 2";P2$
1110 FOR A=1 TO (LEN(P1$)/2)
1120 G1$(A)=MID$(P1$,(A*(A-1)),2)
1130 NEXT A
1140 FOR B=1 TO (LEN(P2$)/2)
1150 G2$(B)=MID$(P2$,(B*(B-1)),2)
1160 NEXT B
1170 G=0
1180 FOR C=1 TO (LEN(P1$)/2)
1190 IF LEFT$(G1$(C),1)>RIGHT$(G1$(C),1) THEN G=G+1
1200 NEXT C
1210 PRINT:PRINT "PARENT 1 HAS";INT(2^G);"GAMETE POSSIBILITIES"
1220 GI=0
1230 FOR D=1 TO (LEN(P2$)/2)
1240 IF LEFT$(G2$(D),1)>RIGHT$(G2$(D),1) THEN GI=GI+1
1250 NEXT D
1260 PRINT:PRINT "PARENT 2 HAS";INT(2^GI);"GAMETE POSSIBILITIES"
1270 FOR CLR=1 TO 16
1280 POKE 
1290 A$(CLR,
1300 B$(CLR,OR)=CHR$(32)
1310 
1320 NEXT CLR
1325 REM----PARENT 1 GAMETES
1330 
1340 
1350 
1360 
1370 
1380 
1390 
1400 FOR F=(LEN(P1$)/2) TO 1 STEP -1
1420 IF LEFT$(G1$(F),1)=RIGHT$(G1$(F),1) THEN 1490
1430 
1440 A$(F1,F)=LEFT$(G1$(F),1)
1450 NEXT F1
1460 FOR F1=(1+2^F2) TO 2^G STEP (2^(F2+1))
1470 A$(F1,F)=RIGHT$(G1$(F),1)
1480 
1490 
1500 
1510 
1520 
1530 IF A$(G1,G2)=CHR$(32) THEN A$(G1,G2)=A$(G1-1),G2)
1540 
1550 NEXT 
1560 
1570 FOR H=1 TO (LEN(P2$)/2)
1580 IF LEFT$(G2$(H),1)>RIGHT$(G2$(H),1) THEN 1610
1590 FOR HI=1 TO (2^GI)
1600 
1610 
1620 
1630 FOR I=(LEN(P2$)/2)  TO 1 STEP -1
1640 IF LEFT$(G2$(I),1)=RIGHT$(G2$(I),1) THEN 1720
1650 F2=F2+1
1660 FOR II=1 TO (2^GI) STEP (2^F2)
1670 B$(II,
1680 NEXT II
1690 FOR II=(1+2^F2) TO 2^GI STEP (2^(F2+1))
1700 B$(II,I)=RIGHT$(G2$(I),1)
1710 NEXT II
1720 NEXT 
1730 IF 2^GI=1 THEN 1790
1740 FOR J1=2 TO 2^GI
1750 FOR J=1 TO (LEN(P2$)/2)
1760 IF B$(J1,J)=CHR$(32) THEN B$(J1,J)=B$(J1-1),J)
1770 NEXT J
1780 NEXT J1
1785 REM----GENERATE OFFSPRING
1790 FOR L=1 TO 128
1800 OS$(L)=CHR$(32)
1810 NEXT L
1830 
1840 FOR X=1 TO 2^G
1850 FOR Z=1 TO (2^GI)
1860 OS$(W)=A$(X,1)+B$(Z,1)+A$(X,2)+B$(Z,2)+A$(X,3)+B$(Z,3)+A$(X,4)
1870 OS$(W)=OS$(W)+B$(Z,4)+A$(X,5)+B$(Z,5)+A$(X,6)+B$(Z,6)
1880 NEXT Z
1890 NEXT X
1900 REM----OUTPUT
```

```
1905 POKE 517,255
1910 PRINT:PRINT "GAMETES OF PARENT 1:":PRINT
1920 FOR GP=1 TO 2^G
1925 PRINT SPC(20);
1930 FOR GU=1 TO (LEN(P1$)/2)
1940 PRINT A$(GP,GU);
1945 GOSUB 4000
1950 NEXT GU
1960 PRINT
1970 NEXT GP
1980 PRINT:PRINT "GAMETES OF PARENT 2:":PRINT
1990 FOR GP=1 TO 2^GI
1995 PRINT SPC(20);
2000 FOR GU=1 TO (LEN(P2$)/2)
2010 PRINT B$(GP,GU);
2015 GOSUB 4000
2020 NEXT GU
2030 PRINT
2040 NEXT GP
2050 PRINT:PRINT "OFFSPRING ——":PRINT
2060 FOR WY=1 TO ((2^G)*(2^GI)) STEP 2
2070 PRINT SPC(5); OS$(WY),OS$(WY+1)
2075 GOSUB 4000
2080 PRINT
2090 NEXT WY
2100 PRINT:PRINT " PERTINENT STATISTICS SUBPROGRAM — '##' CANCEL";
2105 PRINT "S SUBPROGRAM"
2110 PRINT:PRINT "GENE(S)     TIMES OCCURING        %"
2120 PRINT:INPUT GN$
2130 IF LEFT$(GN$,1)="#" THEN 2220
2140 ON LEN(GN$)/2 GOSUB 5000,6000
2200 PRINT SPC(19) CC SPC(12) ((CC/((2^G)*(2^GI)))*100)
2210 GOTO 2120
2220 POKE 517,0
2225 PRINT:PRINT:INPUT "IS ANOTHER CROSS TO BE MADE";YN$
2230 IF LEFT$(YN$,1)="Y" THEN 1000
2240 GOTO 7000
4000 REM----SCROLL CONTROL
4010 IF PEEK(57088)=32 THEN 4030
4020 RETURN
4030 FOR VV=1 TO 200: NEXT VV
4040 IF PEEK(57088)<>32 THEN 4040
4050 FOR VV=1 TO 100:NEXT VV
4060 RETURN
5000 REM——SINGLE GENE COUNT
5010 CC=0
5020 FOR W=1 TO ((2^G)*(2^GI))
5030 FOR WW=1 TO LEN(OS$(1)) STEP 2
5040 IF MID$(OS$(W),WW,2)=GN$ THEN CC=CC+1: GOTO 5070
5050 IF MID$(OS$(W),(WW+1),1)<>LEFT$(GN$,1) THEN 5070
5060 IF MID$(OS$(W),WW,1)=RIGHT$(GN$,1) THEN CC=CC+1
5070 NEXT WW
5080 NEXT W
5090 RETURN
6000 REM----DUAL GENE COUNT
6010 CC=0
6020 FOR W=1 TO ((2^G)*(2^GI))
6030 FOR WW=1 TO LEN(OS$(1)) STEP 2
6040 IF MID$(OS$(W),WW,2)=LEFT$(GN$,2) THEN 6070
6050 IF MID$(OS$(W),(WW+1),1)<>LEFT$(GN$,1) THEN 6100
6060 IF MID$(OS$(W),WW,1)<>MID$(GN$,2,1) THEN 6100
6070 FOR WW=1 TO LEN(OS$(1)) STEP 2
6075 IF MID$(OS$(W),WW,2)=RIGHT$(GN$,2)THENCC=CC+1:GOTO6100
6080 IF MID$(OS$(W),(WW+1),1)<>MID$(GN$,3,1) THEN 6100
6090 IF MID$(OS$(W),WW,1)=RIGHT$(GN$,1) THEN CC=CC+1
6100 NEXT WW
6110 NEXT W
6120 RETURN
7000 END
```

```
OFFSPRING ——

     BBTTXX          BBTtXX
     BbTTXX          BbTtXX
     BBTTYX          BBTtYX
     BbTTYX          BbTtYX
     bBTTXX          bBTtXX
     bbTTXX          bbTtXX
     bBTTYX          bBTtYX
     bbTTYX          bbTtYX

 PERTINENT STATISTICS SUBPROGRAM — '##' CANCELS SUBPROGRAM

GENE(S)     TIMES OCCURING        %

? BBTT                2           12.5

? bbXY                2           12.5

? Tt                  8           50

? ##

          ** GENETICS PROGRAM **

     OFFSPRING GENERATION AND CORRESPONDING STATISTICS

          ———

 SHIFT LOCK MUST BE RELEASED.  USE LEFT SHIFT TO ENTER CAPITALS

          ———

 DURING LISTINGS, SCROLL MAY BE STOPPED / STARTED USING 'ESC' KEY

          ———

ENTER PARENT 1? BbTTXY
ENTER PARENT 2? BbTtXX
PARENT 1 HAS 4 GAMETE POSSIBILITIES

PARENT 2 HAS 4 GAMETE POSSIBILITIES
GAMETES OF PARENT 1:

          BTX
          BTY
          bTX
          bTY

GAMETES OF PARENT 2:

          BTX
          BtX
          bTX
          btX
```

USER - INPUT

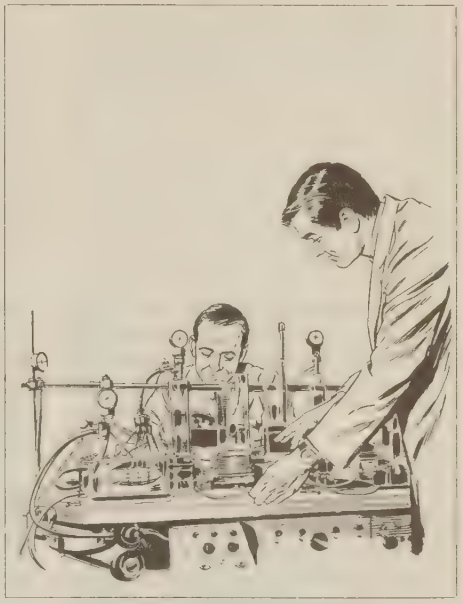
COMPUTER OUTPUT

COMPUTER OUTPUT

(THE REAL TIME-SAVER)

Genetic Engineering

Dick Straw



Have you ever thought you would like to be a genetic engineer? You know, someone with the skills and techniques to make plants and animals to your own designs? Or to eliminate some wasteful genetic disease by skillfully manipulating the hereditary material?

The means for accomplishing real genetic engineering are still not available, although biologists working in that field get closer and closer each year. With the CODON program that follows, you can simulate some of the hoped-for techniques, however, and get some ideas about how the genes do their work. If you happen to be a biology teacher, you can use the program to help students learn some of the basics of molecular biology as well.

The means for accomplishing real genetic engineering are still not available, although biologists get closer and closer each year.

Even if you are just an ordinary programming buff without much concern for genetic engineering or molecular biology, the CODON program might still be of interest to you because of the string-handling routines involved. They show off some important ways that the Basic language is, I feel, superior to Fortran, which handles strings only with difficulty. (Calm the hackles, friend — I use Fortran a lot too, when its advantages are needed!)

Richard M. Straw, 2100 Maiden Lane, Altadena, CA 91001.

In addition, there is a segment that will direct all input and output to the printer without modifying the program to use LPRINT statements. (I owe this idea to Charles Butler, from an article in *Personal Computing.)*

This is not the place for an extended lesson in molecular biology. Almost all recent high school and college general biology books will give a good and more complete description of the details. A book called *Invitation to Biology* by Helena Curtis, used widely for college biology courses taken by non-specialists, is a good example. For a more detailed and superbly written account, try James Watson's (of double helix fame) book, *The Molecular Biology of the Gene.* Still, a very brief account might make the program more interesting.

Practically all the work of the cell is accomplished by proteins. These proteins are large and complex molecules made up of twenty elementary building blocks called amino acids. The numbers, kinds and order of the amino acids determine the shape and thus the function of the protein. Think of them, if you will, as factory "jigs" that hold things together so a needed chemical operation can be carried out. What molecules the proteins hold and what they do with them is determined entirely by the arrangement of their amino acids. Whether the cell or individual survives depends on what they do and how well, however.

What kinds of jigs the cellular factory makes is determined by the kinds of genes in the nucleus of the cell. You can think of genes as the specifications for building jigs. Every time a new factory (cell) is built, the jig-specifying genes are copied and passed on to it from the previous cell. Every time a new individual (a new person or flower, full of cell-factories) is produced, the cells in it get a portion of the genes from each parent and can make new combinations of jigs. This determines the individual's unique characteristics. But the genes are not used directly — they are first copied into "blueprints" to be used in the factory itself, outside the nucleus, where the genes are never allowed to go.

Three are terminators and give a STOP message that cuts off the protein building at that point.

The genes are really coded structures made up of four kinds of molecules called "bases," put together into a complex structure called a nucleic acid. A certain length of the kind of nucleic acid found in chromosomes, called DNA, is a single gene, and it carries the specifications for a single protein to be built. The blueprints are copies of this genetic information in a second kind of nucleic acid, RNA, which also carries its message in the

arrangements of four sub-units. These sub-unit bases are the chemical molecules called adenine, cytosine, guanine, and uracil, and are referred to by their initials. Remember that their order is a code that specifies the organization of a particular protein with a specific function.

When the blueprint-RNA is shipped out into the cell's factory, it is operated on by a cellular structure that reads the message in a very special manner, much as a film-strip projector shows the film one frame at a time. The frames in this case contain three of the bases at a time, and each set of three bases in a row is called a codon — the smallest piece of information in the RNA message. Recall that there are twenty kinds of amino acids but only four kinds of bases. This means that the smallest quantity of coded information that can translate a code of four symbols into another of twenty symbols must have a minimum of three symbols per unit. (There is an exercise in combinatorics for you.) Each codon of the RNA is thus equivalent to one amino acid (except for three that are terminators and give a STOP message that cuts off the protein building at that point). With four symbols taken three at a time in specific order, there are 64 kinds of codons. Since there are only twenty amino acids, you get a lot of synonyms (called redundancy), but that has its advantages too.

Mutations are changes in the structure of the genes that, by way of the messenger RNA, are translated into incorrectly built proteins. These "bad" proteins usually either do something wrong or do not do whatever they were supposed to do. Either of these can upset the functioning of the cell and the individual. Thus things that cause mutations, like radiation and many kinds of chemicals, are bad for the health also.

Enough of that. Let's look at what the program does.

Briefly, CODON invites you to enter a sequence of base initials, A, C, G or U, in any order you desire up to a total of 150 in up to three lines. Short examples do as well as long ones, usually. This message is then echoed back to you in codon-groups, three letters at a time, as shown in Figure 1.

```
RNA MESSAGE NOW READS (BY CODONS):

AUC AUC AUC AUC AUC AUC

YOUR INPUT STRING TRANSLATES TO:

ILE-ILE-ILE-ILE-ILE-ILE-
```

Figure 1. The original input string of RNA bases is echoed in codon groups of three bases and translated into the corresponding amino acid string. Only one codon type is used to make the changes more evident.

Here, a simple sequence repeating the codon AUC was used. The program then translates that RNA message into the corresponding sequence of amino acids it would produce in the cell. In the example, AUC translates into a sequence of the amino acid, isoleucine, abbreviated by the letters ILE. If you look at lines 7160-7250 you will see the abbreviations used, which will be printed for you if you ask for information at the beginning of the program. See the full sample run, too.

You have three options: you can change one base in the message to another, or insert some more bases into the list, or take out some bases.

You then have three options (other than stopping): you can change one base in the message to another, or insert some more bases into the list, or take out some bases. Figure 2 shows what happens when you change base number 8 from U to C. One amino acid in the structure is changed this time, from ILE to THR, or threonine. This simple mutation, representative of the smallest genetic change possible, can be extremely important. The only difference between the normal hemoglobin that most of us have in our cells and the sickle-cell hemoglobin that can kill its carriers through severe anemia is just such a single amino acid substitution, putting in valine instead of the glutamic acid that is usually present. The RNA message change was probably putting a U where an A belonged.

```
RNA MESSAGE NOW READS (BY CODONS):
          ↓
AUC AUC ACC AUC AUC AUC

PROTEIN HAS NOW BECOME:

ILE-ILE-THR-ILE-ILE-ILE-
```

Figure 2. The base at position 8 (indicated by an arrow) was changed from U to C, and the resulting amino acid string has one different acid because of this mutation.

Figure 3 shows what happens when you add a single base. If you examine the sequence, you will find that a new C has been stuck into the sequence at position 7. The list is now one base longer, but the program does not translate the stray end. Two things will be apparent. First, you can see that the whole protein after the insertion has been changed — to a sequence of histidines (HIS). Secondly, the third and fourth codons, CAC and CAU, both translate into HIS, an example of the code redundancy.

```
RNA MESSAGE NOW READS (BY CODONS):
          ↓
AUC AUC CAC CAU CAU CAU C

CURRENT PROTEIN READS:

ILE-ILE-HIS-HIS-HIS-HIS-
```

Figure 3. A new base, C, was inserted into the string after position 6, with the result that the whole message has changed beyond the position of the insertion. Note also that two different codons have been translated into one amino acid — an example of code redundancy.

If you now take out a base, number 13 in the example, you get the results shown in Figure 4. Here you can see

```
RNA MESSAGE NOW READS (BY CODONS):
AUC AUC CAC CAU AUC AUC
          ↑
PROTEIN NOW HAS THIS STRUCTURE:

ILE-ILE-HIS-HIS-ILE-ILE-
```

Figure 4. The arrow indicates the position (#13 of the previous one) from which a base, C, was removed. This causes the amino acid chain to be different between the two changes made, compared to Figure 1, but the same both before and after them.

that there is a place in the sequence between the insertion and the deletion that is different from the original, but both ends are back to where you started. It was by using techniques like this that Nobel prizewinner Francis Crick demonstrated that codons really were three bases long — adding three or taking out three bases leaves the protein between modified but the ends the same. If the modified length is not too great, the protein may still function partly — depending on a lot of things. In general, however, any mutations cause problems that may vary from insignificant to great.

Each set of three bases in a row is called a codon — the smallest piece of information in the RNA message. There are twenty kinds of amino acids but only four kinds of bases.

Each of the parts of the program is marked by REM lines so that it should be easy to follow. The codon dictionary (found in the data lines 6000 to 6080) is read into array C by lines 150 — 170. It is used to translate the RNA message. If your version of Basic does not accept three-dimensional arrays some rewriting will be needed here. Notice also that I used the DEFSTR (define string) and DEFINT (define integer) functions of the TRS-80 Basic to declare the types of several variables ahead of time. At least the C's, A's and S's in the program would need to be changed to C$'s, etc., if you do not have this capacity. Also permitted here, but perhaps not by your interpreter, is the reading of string data without quotation marks from lines 6000 - 6080.

The program segment in lines 211 to 216 allows you to divert all input and output to the printer if you wish. If you answer "yes" to the first question, the two pairs of PEEK statements pick up the addresses of the video driver (VL% and VH% — low and high bytes) and printer driver (PL% and PH%), and then two POKEs put the printer driver address where the video driver address usually resides. When you stop the run (at lines 510-530, by selecting option 4 from the menu), the video addresses are restored to their proper places. The complete run example was produced by this option. It is best not to stop the program with the <BREAK> key.

The routine at lines 340-390 reads in your input string and puts together a long one, C1. The echo routine at subroutine 8700 puts the input string C1 into string C5 and lists the input string by codons. String C5 has spaces inserted between the groups of three bases, which are not wanted by the translating routines.

Translating is done by the subroutine beginning at 8000, which in turn calls another at 8200. The second translates the letters of each successive codon in your input string into numbers by which to access the array C, and the prime subroutine builds another string, A, which is then printed out by the subroutine at 8500. In both printing subroutines line lengths are controlled so a uniform and neat output is produced. The whole program is written for video display; but, as noted above, output can be sent to the printer if you wish.

The segments that insert or delete strings of bases into the original (or any subsequent) string start at lines 2000 and 3000. All they do is find out what location is desired, break the original string into the two ends needed, and put in or take out what you request. If you ask to delete beyond the end of the string, the end is cut off and the extra "deletion" is ignored. You can't delete more than the whole string, though.

The program is written in Disk Basic, but the only unique feature is used in line 1060 of the change-a-base routine. This is the MID$ function on the left side of an assignment statement. To convert to Level II Basic this line would need to be rewritten to use the techniques of the insert or delete methods. The following substitution for line 1060 will work:

```
1060 C2=LEFT$(C1,K1-1):
C3=RIGHT$(C1,L-K1):
C1=C2+A5+C3
```

It is the only change that should be needed, however, as everything else is standard usage.

With that amount of description, everything else should be easily followed. Most common mistakes (like entering an incorrect letter) are trapped, and lots of prompts and information are provided to make this an essentially independent program once its basic intentions are understood.

Figure 5. A full run of the program using the printer option, including the information that is optionally provided.

```
DO YOU WANT INFORMATION AND INSTRUCTIONS? Y

MESSENGER RNA IS A COPY OF THE GENETIC INFORMATION IN
THEN DNA OF THE GENE IN THE CHROMOSOME. THE ORDER OF THE
FOUR BASES (A = ADENINE, C = CYTOSINE, G = GUANINE, AND
U = URACIL) DETERMINES THE ORDER OF THE AMINO ACIDS IN
THE PROTEIN.  THAT ORDER REGULATES THE FUNCTION OF THE
PROTEIN IN THE CELL.

YOU MAY ENTER A STRING OF BASES TO SIMULATE THE
STRUCTURE OF THE RNA MOLECULE.  USE ONLY THE LETTERS,
A, C, G, AND U, AND LEAVE NO SPACES. IF YOU WANT MORE
THAN ABOUT 50, USE 2 OR 3 LINES BUT DO NOT ENTER MORE
THAN 150 BASES IN ALL.

    DO YOU WANT MORE INFORMATION? Y

EACH GROUP OF THREE BASES IN THE MESSAGE IS CALLED A CODON.
EACH CODON TRANSLATES INTO ONE AMINO ACID IN THE PROTEIN.
THE AMINO ACID ABBREVIATIONS ARE:
ALA = ALANINE            LEU = LEUCINE
ARG = ARGENINE           LYS = LYSINE
ASN = ASPARAGINE         MET = METHIONINE
ASP = ASPARTIC ACID      PHE = PHENYLALANINE
CYS = CYSTEINE           PRO = PROLINE
GLN = GLUTAMINE          SER = SERINE
GLU = GLUTAMIC ACID      THR = THREONINE
GLY = GLYCINE            TRP = TRYPTOPHANE
HIS = HISTIDINE          TYR = TYROSINE
ILE = ISOLEUCINE         VAL = VALINE
    READY? Y

YOU MAY NOW ENTER AN RNA MESSAGE.
ENTER NO MORE THAN THREE LINES OF DATA NOR MORE THAN 150
BASES IN ALL, USING ONLY LETTERS A, C, G, AND U
RESPOND TO '?' WITH <ENTER> IF DONE WITH INPUT.
? ACCGUGCACUAGGUCAAUCG
?
LENGTH IS 20

RNA MESSAGE NOW READS (BY CODONS):

ACC GUG CAC UAG GUC AAU CG

YOUR INPUT STRING TRANSLATES TO:

THR-VAL-HIS-STOP

YOU CAN MAKE ANY OF THESE CHANGES:
   1.   CHANGE ONE BASE IN THE LIST.
   2.   INSERT ONE OR MORE ADJACENT BASES.
   3.   DELETE ONE OR MORE ADJACENT BASES.
  OR STOP (TYPE 4)
WHICH DO YOU CHOOSE? 1

CHANGE A SINGLE BASE IN RNA MESSAGE

CHANGE WHICH BASE (NUMBER)? 10
BASE NUMBER 10 IS NOW U

WHAT BASE DO YOU WISH TO CHANGE IT TO? C

RNA MESSAGE NOW READS (BY CODONS):

ACC GUG CAC CAG GUC AAU CG

PROTEIN HAS NOW BECOME:

THR-VAL-HIS-GLN-VAL-ASN-

YOU CAN MAKE ANY OF THESE CHANGES:
   1.   CHANGE ONE BASE IN THE LIST.
   2.   INSERT ONE OR MORE ADJACENT BASES.
   3.   DELETE ONE OR MORE ADJACENT BASES.
  OR STOP (TYPE 4)
WHICH DO YOU CHOOSE? 2

INSERT BASES INTO RNA MESSAGE

ENTER STRING OF BASES TO INSERT:
? CCAAGGUUCAC
INSERT AFTER WHICH BASE (NUMBER)? 18
RNA NOW HAS 31  BASES

RNA MESSAGE NOW READS (BY CODONS):

ACC GUG CAC CAG GUC AAU CCA AGG UUC ACC G

CURRENT PROTEIN READS:

THR-VAL-HIS-GLN-VAL-ASN-PRO-ARG-PHE-THR-

YOU CAN MAKE ANY OF THESE CHANGES:
   1.   CHANGE ONE BASE IN THE LIST.
   2.   INSERT ONE OR MORE ADJACENT BASES.
   3.   DELETE ONE OR MORE ADJACENT BASES.
  OR STOP (TYPE 4)
WHICH DO YOU CHOOSE? 3

DELETE BASES FROM RNA MESSAGE

HOW MANY BASES DO YOU WISH TO REMOVE? 4

DELETE BEGINNING WITH WHICH BASE (NUMBER)? 5
RNA NOW HAS 27  BASES

RNA MESSAGE NOW READS (BY CODONS):

ACC GCC AGG UCA AUC CAA GGU UCA CCG

PROTEIN NOW HAS THIS STRUCTURE:

THR-ALA-ARG-SER-ILE-GLN-GLY-SER-PRO-

YOU CAN MAKE ANY OF THESE CHANGES:
   1.   CHANGE ONE BASE IN THE LIST.
   2.   INSERT ONE OR MORE ADJACENT BASES.
   3.   DELETE ONE OR MORE ADJACENT BASES.
  OR STOP (TYPE 4)
WHICH DO YOU CHOOSE? 4
```

Program Listing

```
10 REM     CODON PROGRAM
20 REM     (C) DICK STRAW 1980
30 REM     TRS-80 DISK BASIC
40 REM
100 CLEAR 2000
110 DEFSTR C,A,S: DEFINT I-N
120 DIM C(4,4,4)
150 FOR I=1 TO 4:FOR J=1 TO 4: FOR K=1 TO 4
160 READ C(I,J,K)
170 NEXT K,J,I
200 CLS:PRINTTAB(20);"CODON PROGRAM"
210 PRINT:PRINT"SIMULATES TRANSLATION OF MESSENGER-RNA INTO PROTEIN"
211 PR%=0: INPUT"DO YOU WANT OUTPUT TO PRINTER";S1
212 IF ASC(S1) <> 89 GOTO 220
213 PR%=1: INPUT"WHEN PRINTER IS READY, PRESS ENTER";S1
214 PL%= PEEK(16422): PH%= PEEK(16423)
215 VL%= PEEK(16414): VH%= PEEK(16415)
216 POKE 16414,PL%: POKE 16415,PH%
220 PRINT:INPUT"DO YOU WANT INFORMATION AND INSTRUCTIONS";S1
230 IF ASC(S1)=89 THEN GOSUB 7000
299 REM    START MAIN PROGRAM
300 PRINT:PRINT"YOU MAY NOW ENTER AN RNA MESSAGE."
310 PRINT"ENTER NO MORE THAN THREE LINES OF DATA NOR MORE THAN 150"
320 PRINT"BASES IN ALL, USING ONLY LETTERS A, C, G, AND U"
330 PRINT"RESPOND TO '?' WITH <ENTER> IF DONE WITH INPUT."
340 C1=""
350 FOR I=1 TO 3
360 C2="": INPUT C2
370 IF C2="" GOTO 400
380 C1=C1+C2
390 NEXT I
400 L=LEN(C1): PRINT"LENGTH IS";L
405 GOSUB 8700
410 PRINT:PRINT"YOUR INPUT STRING TRANSLATES TO:"
420 GOSUB 8000        'TRANSLATING ROUTINE
430 GOSUB 8500        'PRINT PROTEIN
440 PRINT:PRINT"YOU CAN MAKE ANY OF THESE CHANGES:"
450 PRINT"   1.   CHANGE ONE BASE IN THE LIST."
460 PRINT"   2.   INSERT ONE OR MORE ADJACENT BASES."
470 PRINT"   3.   DELETE ONE OR MORE ADJACENT BASES."
480 PRINT"  OR STOP (TYPE 4)"
490 INPUT"WHICH DO YOU CHOOSE";N
500 ON N GOTO 1000, 2000, 3000, 510
510 IF PR%=0 GOTO 530
520 POKE 16414,VL%: POKE 16415,VH%
530 STOP
999 REM  CHANGE ONE BASE
1000 PRINT:PRINT"CHANGE A SINGLE BASE IN RNA MESSAGE"
1010 PRINT:INPUT"CHANGE WHICH BASE (NUMBER)";K1
1020 IF K1>L OR K1<0 PRINT"INVALID NUMBER":GOTO 1010
1030 A5= MID$(C1,K1,1)
1040 PRINT"BASE NUMBER";K1;"IS NOW ";A5
1050 INPUT"WHAT BASE DO YOU WISH TO CHANGE IT TO";A5
1060 MID$(C1,K1,1) = A5
1070 GOSUB 8700          'PRINT RNA
1080 PRINT:PRINT"PROTEIN HAS NOW BECOME:"
1090 GOSUB 8000
1100 GOSUB 8500
1110 GOTO 440
1999 REM  INSERT BASES
2000 PRINT:PRINT"INSERT BASES INTO RNA MESSAGE"
2010 PRINT: A5="": PRINT"ENTER STRING OF BASES TO INSERT:"
2020 INPUT A5
2060 L2=LEN(A5): IF L2+L > 150 PRINT"TOO LONG":GOTO 2010
2070 INPUT"INSERT AFTER WHICH BASE (NUMBER)";K1
2080 IF K1>L PRINT"NOT A VALID LOCATION - PAST END":GOTO 2070
2090 C2=LEFT$(C1,K1): L2=L-K1
2100 C3=RIGHT$(C1,L2)
2120 C1=C2+A5+C3: L=LEN(C1)
2130 PRINT"RNA NOW HAS";L;" BASES"
2140 GOSUB 8700
2150 PRINT:PRINT"CURRENT PROTEIN READS:"
2160 GOSUB 8000
2170 GOSUB 8500
2180 GOTO 440
2999 REM   REMOVE BASES
3000 PRINT:PRINT"DELETE BASES FROM RNA MESSAGE"
3010 PRINT:INPUT"HOW MANY BASES DO YOU WISH TO REMOVE";L2
3020 IF L-L2 < 0 PRINT"RNA IS NOT THAT LONG":GOTO 3010
3030 PRINT:INPUT"DELETE BEGINNING WITH WHICH BASE (NUMBER)";K1
3040 L3=L-K1-L2: IF L3<0 THEN L2=L-K1: GOTO 3040
3050 C2=LEFT$(C1,K1-1): C3=RIGHT$(C1,L3+1)
3060 C1=C2+C3: L=LEN(C1)
3070 PRINT"RNA NOW HAS";L;" BASES"
3080 GOSUB 8700
3090 PRINT:PRINT"PROTEIN NOW HAS THIS STRUCTURE:"
3100 GOSUB 8000
3110 GOSUB 8500
3120 GOTO 440
4000 END
4001 REM

6000 REM   CODON DICTIONARY
6010 DATA PHE-,PHE-,LEU-,LEU-,SER-,SER-,SER-,SER-
6020 DATA TYR-,TYR-,STOP,STOP,CYS-,CYS-,STOP,TRP-
6030 DATA LEU-,LEU-,LEU-,LEU-,PRO-,PRO-,PRO-,PRO-
6040 DATA HIS-,HIS-,GLN-,GLN-,ARG-,ARG-,ARG-,ARG-
6050 DATA ILE-,ILE-,ILE-,MET-,THR-,THR-,THR-,THR-
6060 DATA ASN-,ASN-,LYS-,LYS-,SER-,SER-,ARG-,ARG-
6070 DATA VAL-,VAL-,VAL-,VAL-,ALA-,ALA-,ALA-,ALA-
6080 DATA ASP-,ASP-,GLU-,GLU-,GLY-,GLY-,GLY-,GLY-
6999 REM
7000 REM  INFORMATION TEXT
7010 PRINT:PRINT"MESSENGER RNA IS A COPY OF THE GENETIC INFORMATION IN"
7020 PRINT"THEN DNA OF THE GENE IN THE CHROMOSOME. THE ORDER OF THE"
7030 PRINT"FOUR BASES (A = ADENINE, C = CYTOSINE, G = GUANINE, AND"
7040 PRINT"U = URACIL) DETERMINES THE ORDER OF THE AMINO ACIDS IN"
7050 PRINT"THE PROTEIN.   THAT ORDER REGULATES THE FUNCTION OF THE"
7060 PRINT"PROTEIN IN THE CELL."
7070 PRINT:PRINT"YOU MAY ENTER A STRING OF BASES TO SIMULATE THE"
7080 PRINT"STRUCTURE OF THE RNA MOLECULE.  USE ONLY THE LETTERS,"
7090 PRINT"A, C, G, AND U, AND LEAVE NO SPACES. IF YOU WANT MORE"
7100 PRINT"THAN ABOUT 50, USE 2 OR 3 LINES BUT DO NOT ENTER MORE"
7110 PRINT"THAN 150 BASES IN ALL."
7120 PRINT:INPUT"    DO YOU WANT MORE INFORMATION";S1
7130 IF ASC(S1) <> 89 THEN RETURN
7140 PRINT:PRINT"EACH GROUP OF THREE BASES IN THE MESSAGE IS CALLED A CODON."
7150 PRINT"EACH CODON TRANSLATES INTO ONE AMINO ACID IN THE PROTEIN."
7155 PRINT"THE AMINO ACID ABBREVIATIONS ARE:"
7160 PRINT"ALA = ALANINE            LEU = LEUCINE"
7170 PRINT"ARG = ARGENINE           LYS = LYSINE"
7180 PRINT"ASN = ASPARAGINE         MET = METHIONINE"
7190 PRINT"ASP = ASPARTIC ACID      PHE = PHENYLALANINE"
7200 PRINT"CYS = CYSTEINE           PRO = PROLINE"
7210 PRINT"GLN = GLUTAMINE          SER = SERINE"
7220 PRINT"GLU = GLUTAMIC ACID      THR = THREONINE"
7230 PRINT"GLY = GLYCINE            TRP = TRYPTOPHANE"
7240 PRINT"HIS = HISTIDINE          TYR = TYROSINE"
7250 PRINT"ILE = ISOLEUCINE         VAL = VALINE"
7260 INPUT"    READY";S1
7270 RETURN
7999 REM
8000 REM  TRANSLATING ROUTINE
8010 A="": FOR I=1 TO L STEP 3
8020 C3="": C3= MID$(C1,I,3)
8025 IF LEN(C3) < 3 GOTO 8130
8030 A2 = LEFT$(C3,1): GOSUB 8200
8040 J1=J: IF J=0 GOTO 8120
8050 A2 = MID$(C3,2,1): GOSUB 8200
8060 J2=J: IF J=0 GOTO 8120
8070 A2 = RIGHT$(C3,1): GOSUB 8200
8080 J3=J: IF J=0 GOTO 8120
8090 A1= C(J1,J2,J3): A=A+A1: IF A1="STOP" GOTO 8130
8100 NEXT I
8120 IF J=0 PRINT"BAD BASE IN CODON AT POSITION";I;" - ";C3
8130 RETURN
8200 REM DECODING BASES TO DICTIONARY INDICES
8205 J=0
8210 IF A2="U" THEN J=1: GOTO 8250
8220 IF A2="C" THEN J=2: GOTO 8250
8230 IF A2="A" THEN J=3: GOTO 8250
8240 IF A2="G" THEN J=4
8250 RETURN
8500 REM    PRINT PROTEIN
8510 IF LEN(A) < 57 PRINT :PRINT A: RETURN
8520 A1 = LEFT$(A,56)
8530 A2 = MID$(A,57,56): A3="": A4=""
8540 IF LEN(A) > 112 THEN A3=MID$(A,113,56)
8550 IF LEN(A) > 168 THEN A4=MID$(A,169,56)
8560 PRINT:PRINT A1: PRINT: PRINT A2
8570 IF LEN(A3) > 0 THEN PRINT: PRINT A3
8580 IF LEN(A4) > 0 THEN PRINT: PRINT A4
8590 RETURN
8700 REM    PRINT RNA MESSAGE
8705 PRINT:PRINT"RNA MESSAGE NOW READS (BY CODONS):"
8710 C5="": FOR I=1 TO LEN(C1) STEP 3
8720 C5=C5+MID$(C1,I,3)+" "
8725 NEXT I
8730 IF LEN(C5)<57 PRINT: PRINT C5: RETURN
8740 C2=LEFT$(C5,56): C3=MID$(C5,57,56): C7="":C8=""
8750 IF LEN(C5) > 112 THEN C7=MID$(C5,113,56)
8760 IF LEN(C5) > 168 THEN C8=MID$(C5,169,56)
8770 PRINT:PRINT C2: PRINT: PRINT C3
8780 IF LEN(C7)>0 THEN PRINT:PRINT C7
8790 IF LEN(C8)>0 THEN PRINT:PRINT C8
8800 RETURN
```

A Problem Solving Technique Based on Genetics

Arthur Radcliffe

"Artificial intelligence," that ever-farther goal, may be a will-o-the-wisp; yet its many pathways of research have brought new insights to the study of psychology, language, and even, as described here, genetics.

Evolution is a process of multidimensional search: each newborn creature is exploring another position in genetic survival-space. Now artificial intelligence research has discovered that one of the best methods of multidimensional search is what we have always called sex. *There being two sexes may not be merely an accidental strategy of protoplasm, but the most efficient scheme of search and adaptation. — TN*

Mother Nature, in her wisdom, comes up with remarkable solutions to the problem of filling all available ecological niches on this planet; and perhaps elsewhere too. Beneath the surface charm of such diverse entities as butterflies and dolphins must therefore lie subtle and powerful evolutionary methodologies. Bypassing consideration of Who brought all this to pass, let us simply borrow a scheme for whose efficacy there is more than sufficient empirical evidence.

Curled in the nuclei of each cell of all the diverse creatures about us are strings of beads, each bead a part of the answer as to how to build and maintain the particular creature of which it is a part. (Note: the word "strings" in the preceding sentence is a clue to where I am leading you; keep it in mind.) If you want to know more about the chemistry of all this, read up on chromosomes and genes and DNA and double helices.

Remaining biological for a while, I wish to raise the topic, rare in computer literature, of sex. For an essential aspect of the proliferation of the species is sexual reproduction. In case you have forgoten, sex facilitates the mingling and passing on of minor variations of a main theme—offspring end up with half the beads of each parent.

Now we come to the vital process of selection; whether natural or un-natural, selection leads to changes in some favorable dimension, perhaps at the expense of other dimensions which may never be explored.

Beneath the surface charm of such diverse entities as butterflies and dolphins must lie subtle and powerful evolutionary methodologies.

Selection is based on some concept of fitness, of greater merit in a given environment. It may not make a turkey happier to have twice the breast meat, but the breeder smiles all the way home from the bank.

So let us focus on the concept of solving a problem by stringing together solution parameters, something easily done on a computer. Solution parameters are whatever elements we are searching for. Thus, we might form a string of the numbers which represent the temperatures, pressures and flow rates in an oil refinery, or string together symbols defining a musical tune. The point is that this method works for problems whose answers can be expressed in a string of symbols, not necessarily numbers.

Given that we have a problem whose answer can be expressed in string form, we proceed to form a population of strings representing possible answers. We may form this population randomly, with malice aforethought, or using precognition if it is available. Once we have a population of strings, it is time for selection. If does not matter how, but we must list the population in order of preference. We may decide this matter of ranking analytically, by mathematical methods; or we may solve it pragmatically, by weighing results on a scale or scanning a profit statement. Or we can rank the strings of our population interactively, esoterically: "I like that pattern on my display screen better than any of the others!"

Now for the last step in the process. We pair off the higher-ranking strings to generate offspring who will replace the lower-ranking strings, which are consigned to oblivion.

That is it, except for a bit of seasoning, salt and pepper in the form of a few random mutations. We just turn the string population over to the proclivities of a computer program which repeatedly executes the above sequence while we have a beer and voyeuristically watch the answer to the problem evolve.

To illustrate the genetic adaptive approach, a problem was written which employs binary strings. These strings define the X and Y coordinates of a point on the display screen. The merit of a given string is determined by the distance of its point from a target point. The odd positions along the string contain the bits of a binary number for one coordinate and the even positions contain a binary number for the other coordinate. The strings are created in line 910 and the target is plotted in line 980.

The strings are converted from binary numbers to decimal X,Y coordinates in lines 160 and 170 using a binary weighting

Arthur Radcliffe, 1521 Hatcher Crescent, Ann Arbor, MI 48103.

array created in line 910. Old points are deleted from the screen in line 230 and new points are added in line 280.

The distances of the points from the target are calculated in line 310. These distances are placed in rank order by the sequence from 400 to 490. The array R contains the serial numbers of the strings in the order ranked. Starting at line 500, the strings are copied from array C in to array A in rank order. Lines 520 to 540 introduce random mutations. The distances of the best four from the target are printed in line 620 and these four are copied into the top positions in array C in line 640.

The mating process is accomplished by pairing strings 1 and 2, then strings 3 and 4. Each pair is cut at random position and the head of each spliced to the tail of the other. Line 740 defines the cutting point; the heads are copied in lines 760 and 770; the tails are copied in lines 800 and 810.

At this point we have:

1.) extricated the odd and even bits from the string to define the X and Y coordinates
2.) plotted points defined by these coordinates
3.) calculated the distances of these points from the target
4.) ranked the strings by closeness to the target
5.) mutated the strings and rearranged them in rank order
6.) established the best four at the top
7.) mated the top four by cutting and slicing to create offspring replacing the bottom four.

That is all there is to it:

A) establish a rating process
B) create an initial population of strings
C) record the rating of each string
D) place the strings in rank order
E) mate and mutate the better strings to replace the poorer ones
F) continue the sequence C, D, E.

When are you done? The process tends toward a point of diminishing returns, often after 20 generations. If you do not like what is happening, tinker with the mating and mutation processes.

The remarks appended to the program should be studied as counterpoint to what has been written above. Experimentation is encouraged: for example, if the order of the bits for one coordinate is reversed, the head of each string will contain the most significant bits for one coordinate and the tail will contain the most significant bits of the other coordinate. Also, more kinky mating processes can be designed.

Credit for creating this genetic adaptation process belongs with John Holland of the Computer Science Department of The University of Michigan. □

Bibliography

1) J.H. Holland, *Adaptation in Natural and Artificial Systems*. U. Michigan Press, 1975.

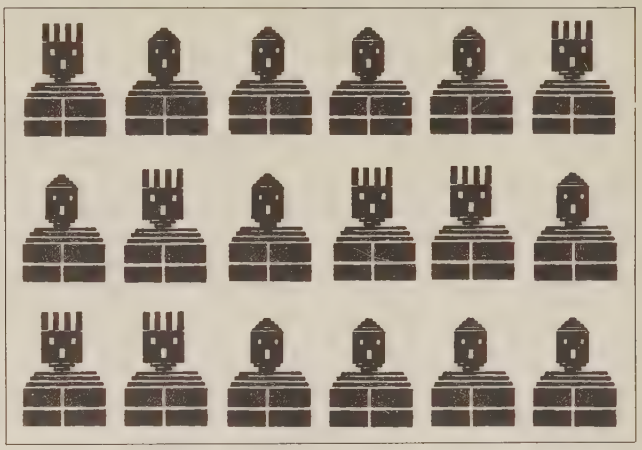
```
1  PRINT : PRINT : PRINT : PRINT "PROBLEM SOLVING USING GENETIC A
   DAPTATION"
10 A = B = C = D = E = F = G = X = Y = 0
20  DIM A(7,13): DIM B(7): DIM C(7,13): DIM D(7): DIM R(7): DIM W
   (7): DIM X(7): DIM Y(7): DIM Z(7)
30  PRINT "CONTACT ART RADCLIFFE                    "
40  PRINT "1521 HATCHER CRESCENT                    "
50  PRINT "ANN ARBOR, MI  48103                     "
60  PRINT "(313)-769-3039                           "
70  PRINT "SEE REMARKS STARTING LINE 1000           "
80  GOTO 900
99  REM  TRANSLATE BINARY STRING TO X & Y COORDINATES:
100  PRINT "GENERATION=";H;
110  FOR D = 0 TO 7:X(D) = 0:Y(D) = 0: NEXT D
120  FOR A = 0 TO 7: FOR B = 0 TO 6
130 C = 2 * B
140 D = C + 1
150 E = B(B)
160 X(A) = X(A) + C(A,C) * E
170 Y(A) = Y(A) + C(A,D) * E
180  NEXT B: NEXT A
199  REM   PLOT:
200  FOR A = 0 TO 7
210 X = W(A)
220 Y = Z(A)
230  HCOLOR= 0: HPLOT X,Y: HPLOT X + 1,Y: HPLOT X,Y + 1: HPLOT X +
   1,Y + 1
240 X = X(A)
250 W(A) = X
260 Y = Y(A)
270 Z(A) = Y
280  HCOLOR= 3: HPLOT X,Y: HPLOT X + 1,Y: HPLOT X,Y + 1: HPLOT X + 1,Y + 1
290  NEXT A
299  REM   CALC. DISTANCES FROM TARGET:
300  FOR A = 0 TO 7
310 D(A) =  INT (.5 +  SQR ((X(A) - 64) ↑ 2 + (Y(A) - 64) ↑ 2))
320 R(A) = A
330  NEXT A
399  REM   ESTABLISH RANK ORDER:
400  FOR A = 0 TO 6: FOR B = A TO 7
410 C = D(A)
420 D = D(B)
430  IF C <  = D THEN 490
440 D(A) = D
450 D(B) = C
460 E = R(A)
470 R(A) = R(B)
480 R(B) = E
490  NEXT B: NEXT A
499  REM   MUTATE AND COPY INTO ARRAY 'A':
500  FOR A = 0 TO 7
510 B = R(A)
520  IF  INT (2 *  RND (1)) = 0 THEN 550
530 D =  INT (14 *  RND (1))
540 C(A,D) = 1 - C(A,D)
550  FOR C = 0 TO 13
560 A(A,C) = C(B,C)
570  NEXT C: NEXT A
599  REM   COPY BEST 4 INTO ARRAY 'C':
600  PRINT "  BEST 4:";
610  FOR A = 0 TO 3
620  PRINT " ";D(A);
630  FOR B = 0 TO 13
640 C(A,B) = A(A,B)
650  NEXT B: NEXT A
660  PRINT
699  REM   CUT & SPLICE 1ST 4 TO MAKE NEW 2ND 4:
700  FOR A = 0 TO 2 STEP 2
710 B = A + 1
720 C = A + 4
730 D = A + 5
740 E = 1 +  INT (12 *  RND (1))
750  FOR F = 0 TO E
760 C(C,F) = A(A,F)
770 C(D,F) = A(B,F)
780  NEXT F
790  FOR G = E + 1 TO 13
800 C(C,F) = A(B,F)
810 C(D,F) = A(A,F)
820  NEXT G: NEXT A
830 H = H + 1
840  IF H = 21 THEN H = 0: GOTO 900
850  GOTO 100
899  REM   INITIALIZE:
900  FOR A = 0 TO 7: FOR B = 0 TO 13
910 C(A,B) =  INT (2 *  RND (1))
920  NEXT B: NEXT A
930  FOR A = 0 TO 6
940 B(A) = 2 ↑ A
```

```
950  NEXT A
960  FOR A = 0 TO 10: PRINT : NEXT A
970  HGR
980  HCOLOR= 3: FOR X = 64 TO 66: FOR Y = 64 TO 66: HPLOT X,Y: NEXT
     Y: NEXT X
990  GOTO 100
1000  REM   THIS IS AN APPLICATION OF A PERFECTLY GENERAL PROBLEM
      SOLVING SYSTEM WHICH I HAVE ALSO USED TO SOLVE THE TRAVELIN
     G SALESMAN PROBLEM: FIND THE SHORTEST ROUTE THROUGH N CITIES
     .
1001  REM   THIS TECHNIQUE WAS DEVELOPED UNDER PROFESSOR JOHN HOL
     LAND OF THE UNIVERSITY OF MICHIGAN COMPUTER SCIENCE DEPARTME
     NT BY D. J. CAVICCHIO AND R. B. HOLLSTIEN ABOUT 1970.
1002  REM   THIS TECHNIQUE CAN BE APPLIED TO ANY PROBLEM WHOSE SO
     LUTION CAN BE EXPPESSED AS A STRING, SUCH AS BY FORMING A ST
     PING FROM THE COEFFICIENTS OF AN EQUATION OR FROM ANY SYMBOL
     S DEFINING A PATTERN OR EVENT.
1003  REM  THERE IS ONE OTHER ESSENTIAL CONDITION: THAT THE POPUL
     ATION OF STRINGS CAN BE RANKED AS TO MERIT (EITHER ANALYTICA
     LLY, BY EXPERIMENT OR BY JUDGEMENT).
1004  REM   THE INITIAL FAMILY OF STRINGS CAN BE BASED ON ESIMATE
      OR CAN BE COMPLETELY RANDOM.
1005  REM   THE PROCEDURE STARTS WITH ASSIGNING A MERIT FACTOR TO
      EACH STRING.
1006  REM   THE NEXT STEP IS TO PLACE THE STRINGS IN ORDER OF MER
     IT.
1007  REM   PAIRS OF STRINGS FROM THE TOP OF THE LST ARE THEN 'MA
     TED' TO PRODUCE OFFSPRING WHICH REPLACE STRINGS AT THE BOTTO
     M OF THE LIST.
1008  REM   THE STRINGS ARE ANALOGOUS TO CHROMOSOMES AND THE PROC
     ESS IS ANALOGOUS TO SELECTIVE BREEDING.
1009  REM   THE MATING CAN CONVENIENTLY BE ACCOMPLISHED BY CUTTIN
     G A PAIR AT RANDOM LOATION AND SPLICING THE HEADS AND TAILS
     TO FORM TWO NEW STRINGS (WHICH EACH CONTAIN ATTRIBUTES OF BO
     TH PARENTS).
1010  REM   IT IS HELPFUL TO THE EVOLUTIONARY PROCESS TO CREATE R
     ANDOM MUTATIONS, AS BY RANDOMLY SELECTING HALF THE POPULATIO
     N FOR ALTERATION OF A RANDOMLY CHOSEN ELEMENT OF THE STRING.

1011  REM   THE POPULATION WITH IT'S NEW MEMBERS AND MUTATIONS IS
      RECYCLED THROUGH THE SAME STEPS, STARTING WITH RATING AND R
     ANKING.  EACH CYCLE IS TERMED A GENERATION.
1012  REM   IN THIS EXAMPLE EIGHT STRINGS OF 14 1'S AND 0'S CONST
     ITUTE THE POPULATION.  THE ODD POSITIONS IN A STRING ARE THE
      BINARY EXPRESSION OF THE X COORDINATE OF A POINT AND THE EV
     EN POSITIONS DEFINE THE Y COORDINATE.
1013  REM   A TARGET IS DEFINED AS THE POINT 64,64.  THE STRINGS
     ARE RATED ACCORDING TO THE DISTANCE OF THE POINT THEY DEFINE
      FROM THIS TARGET.
1014  REM   ARRAY C CONTAINS 8 STRINGS.  ARRAY A IS USED TO BUFFE
     R ARRAY C DURING BREEDING.  ARRAY B CONTAINS THE VALUES OF T
     HE POSITIONS OF A BINARY NUMBER.  ARRAY D CONTAINS THE DISTA
     NCE OF EACH POINT FROM THE TARGET.
1015  REM   ARRAY R CONTAINS THE SERIAL NUMBERS OF THE STRINGS IN
      RANK ORDER.  ARRAYS X AND Y STORE THE X AND Y COORDINATES D
     EFINED BY EACH STRING.  ARRAYS W AND Z BUFFER ARRAYS X AND Y
     .
1016  REM   LINES 0-90: GET ORGANIZED.
1017  REM   LINES 900-990: INITIALIZE.
1018  REM   LINES  100-180: TRANSLATE STRING TO X AND Y COORDINAT
     ES.
1019  REM   LINES 200-290: PLOT THE POINTS.
1020  REM   LINES 300-330: CALCULATED DISTANCES OF POINTS FROM TA
     RGET.
1021  REM   LINES 400-490: ESTABLISH RANK ORDER.
1022  REM   LINES 500-570: MUTATE AND COPY STRINGS INTO ARRAY 'A'
      IN RANK ORDER.
1023  REM   LINES 600-660: COPY TOP (BEST) 4 STRINGS INTO ARRAY '
     C'.
1024  REM   LINES 700-850: CUT AND SPLICE COPIES OF THE BEST 4 TO
      REPLACE THE WORST 4.
1025  REM   THISPROGRAM WAS WRITTEN TO ILLUSTRATE THE PROCESS.  S
     ECTION 200 CAN BE MERGED WITH SECTION 100 TO ELIMINATE ARRAY
     S W AND Z, FOR INSTANCE.
1026  REM   IT IS INTERESTING TO EXPLORE POPULATION SIZES, MATING
      PROCESSES AND MUTATION RATES.  20 GENERATIONS SEEMS TO BE A
      MAGIC NUMBER FOR APPROACHING DIMINISHING RETURNS.
1027  REM   THE STRING ELEMENTS COULD BE SYMBOLS CODING A MUSICAL
      PIECE OR OTHER ARTISTIC COMPOSITION WITH THE RANKING BEING
     BASED ON ESTHETIC MERIT.
1028  REM   THE STRING ELEMENTS COULD ALSO REPPESENT CONTROL SETT
     INGS FOR A CHEMICAL PROCESSING PLANT.
1029  REM   THE RANKING COULD BE PROVIDED BY MEASUREMENTS OF PLAN
     T GROWTH OR BY ANIMAL ACTIVITY RECORDED WITH APPROPRIATE SEN
     SORS.
1030  REM   PLEASE CONTACT ART RADCLIFFE IN ANN ARBOR AND SHARE Y
     OUR EXPERIENCE: PROBLEMS AND RESULTS, WITH THIS SYSTEM.  THA
     NX.
]
```

Genepool

J.D. Eisenberg

A Genetics Simulation in Pascal



It all started with Gregor Mendel, who discovered the rules of genetics from his experiments breeding peas. Biology students have, for years, bred fruit flies in the laboratory to see these rules in action. Now, the *Genepool* program allows you to simulate this kind of experiment in the comfort of your own home on an Apple II.

Background

You can use *Genepool* effectively if you have studied some of basic genetic theory:

Humans have *traits* (such as eye color, hair color, etc.) which are passed to their offspring by the *genes* carried in their cells.

Figure 1.

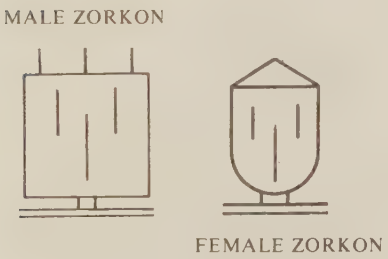
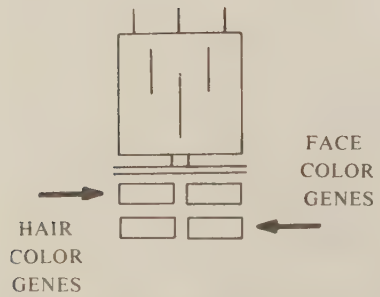


Figure 2.



J.D. Eisenberg, 1630 Ottawa Ct., Apt. K, Sunnyvale, CA 94087.

TABLE I

IF YOU CARRY THESE GENES:		YOUR EYE COLOR WILL BE:
BROWN (DOMINANT)	BROWN (DOMINANT)	BROWN (DOMINANT)
BROWN (DOMINANT)	BLUE (RECESSIVE)	BROWN (DOMINANT)
BLUE (RECESSIVE)	BLUE (RECESSIVE)	BLUE (RECESSIVE)

Eye color, as an example, is determined by genes which are "tagged" for brown eyes or blue eyes. The combination of genes that you received, one from each parent (as detailed in Table I) determines what eye color you have.

In this example, the brown eye color gene is *dominant* over the blue eye color gene; if either gene is for brown eyes, you will have brown eyes. The blue eye gene is called *recessive;* you must have both your eye color genes "tagged" for blue to have blue eyes.

A combination of both dominant genes for a trait is called *homozygous dominant* (see Glossary). A combination of both recessive genes for a trait is called *homozygous recessive*. Finally, the combination of one dominant and one recessive gene for a trait is called *heterozygous dominant.*

Using the Program

The *Genepool* program allows you to see the results of combining genetic types. When you first run the program, you will see pictures of mythical creatures called Zorkons. They may have either green or white hair, and orange or blue faces. The gene for orange faces is dominant over the gene for blue faces, and the gene for green hair is dominant over the gene for white hair.

You can tell the males from the females easily; the males have square faces and straight hair; the females have rounder faces and flat, piled-up hair, as in Figure 1.

Beneath each Zorkon are two rows of boxes, as in Figure 2. The top row shows the hair color genes that the Zorkon carries; the bottom row shows its face color genes.

The Zorkons at the top of the screen are the current genetic types. Beneath them are twenty-four of their possible offspring, created by randomly selecting one gene from each parent for each trait.

At the very start of the program, you may either press the question mark (?) key for a brief explanation of the theory of the program, or you may press the spacebar to start experimenting.

When experimenting, the program will ask you to choose one male genetic type, and one female genetic type.

If your Apple has game paddles, you use paddle zero to move a cursor to the genetic type you wish to choose, and push the button on paddle zero to make the selection.

If you do not have game paddles, you are prompted to use the backward and forward arrow keys to move the cursor, and press the spacebar to make your selection. Once you have selected the male and female genetic types, you will see them

at the top of the screen, and twenty-four of their potential offspring on the screen.

You may, at any time, press any of these keys:

RETURN to start the program over
ESC to exit from the program
S to see statistics

The statistics summarize how many of each genetic combination (homozygous dominant, etc.) there are in the current batch of twenty-four offspring.

Questions to Answer

Try using the program to answer these questions:

1. Is it possible to select genetic types so that you get only one kind of hair color or face color?

2. How many times do you have to choose (given luck of the draw) to get a screenful of Zorkons with only blue faces and green hair? (Once you get this combination, you can press RETURN to start all over again.)

3. Is there a way to assure that all the offspring in a batch will be heterozygous dominant for hair color?

In short, experiment and see what you can find out about genetics from this program. Remember, this program is a simplified simulation, and not a real genetics experiment. If you have interest in finding out more about genetics, consult your local library.

Inside the Program

Here is a brief description of the data structures and procedures inside the program, listed as Program A.

The major data structures that make the program work are in the TYPE definitions. The SEXTYPE, GENETYPE, and CHARACTERISTIC types are meant primarily to make the program easier to read by using symbolic names rather than numbers. To specify an offspring completely within the program, it is necessary to know its sex, what two genes it carries for each trait, and which gene (dominant or recessive) is outwardly visible for each trait. In many other languages, including Basic, it might be necessary to keep these pieces of information in separate variables. Pascal, however, allows you to put all this information in separate variables. Pascal, however, allows you to put all this information together in a single "packet," or RECORD. ZYGOTE is exactly such a record. This allows the program to create an array of twenty-four such "packets" of information (one for each of the displayed offspring) in THEKIDS.

STATS is also an array consisting of a RECORD of three integers that is used to keep statistics on the offspring's genetic types.

COLOURS is an array that tells which colors to display for each trait's dominant and recessive genes.

Here are the procedures and functions used by the *Genepool* program, in order of their appearance:

PROCEDURE SKIPTO (X,Y:INTEGER);

Moves to the specified X,Y screen coordinate; meant mostly as a convenience to avoid typing.

PROCEDURE DRAWFACE (PERSON:ZYGOTE;ATROW, ATCO:INTEGER);

Takes the information about the PERSON and draws its face and hair at the specified "row" and "column" of faces (ATROW,ATCOL) where each "row" holds eight faces.

PROCEDURE DRAWGENES (PERSON:ZYGOTE;ATROW, ATCOL:INTEGER);

Takes information in PERSON and plots the boxes beneath the face. ATROW and ATCOL are as in DRAWFACE.

PROCEDURE DRAWME (PERSON:ZYGOTE;ATROW, ATCOL:INTEGER);

Draws all the information about the PERSON by doing DRAWFACE followed by DRAWGENES. Note: the two procedures are separate because, in the help section, it is necessary to draw the faces independent of the gene boxes.

PROCEDURE AT (COL,ROW:INTEGER);

Moves to text row ROW, column COL on the hi-res page, in preparation for writing text via WCHAR or WSTRING.

PROCEDURE ERSLINE (ROW:INTEGER);

Erases the given ROW on the hi-res screen.

PROCEDURE PSPACEBAR;

Gives prompt to press the spacebar to continue, then waits for either spacebar or the ESCAPE key.

PROCEDURE GENERATION;

This procedure draws the parent genetic types at the top of the screen, then generates a new set of twenty-four offspring. Offspring are generated by randomly choosing one gene for each trait from each parent. Statistics about the offspring are tabulated during this process. The offspring's sex is chosen at random but for two children, one of whom will be male and the other of whom will be female. This assures that you will have at least one female and one male genetic type.

PROCEDURE STATISTICS;

Produces a summary of the traits possessed by the current generation of Zorkons.

PROCEDURE SHOWZYGOTE (WHO:ZYGOTE;WHAT: CHARACTERISTIC);

Takes a zygote (WHO) and tells whether it is homozygous or heterozygous, dominant or recessive for hair or face color characteristic (WHAT). Used in STATISTICS to summarize Mom & Dad's traits.

PROCEDURE LEGEND;

Puts up pictures of Mom and Dad Zorkon, with appropriate labels. Used within the help pages.

PROCEDURE EXPLAIN;

First of the help pages.

PROCEDURE SETCOLOURS (DOMFACE,RECFACE,DOMHAIR, RECHAIR:SCREENCOLOR);

Sets the DOMinant FACE, RECessive FACE, DOMinant HAIR, and RECessive hair colors. (This procedure lets EXPLAIN show a face and no hair by setting hair color to black, for example.)

PROCEDURE EXPLAIN2;
PROCEDURE EXPLAIN3;
PROCEDURE EXPLAIN4;

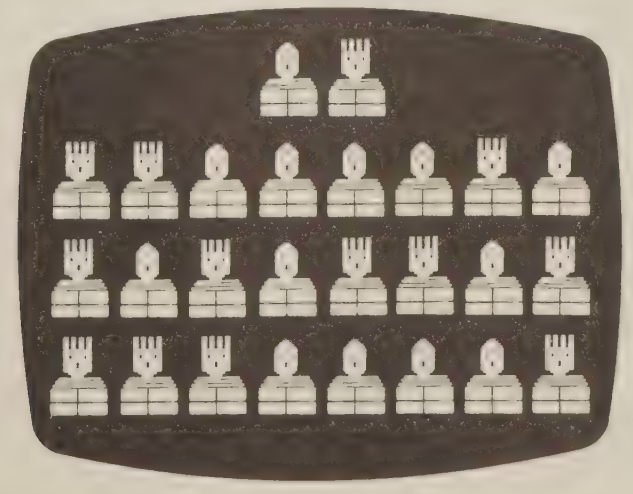
The remainder of the help pages.

PROCEDURE SPLASHPAGE;

Main title page; the help pages are called from here.

PROCEDURE INITPARENTS;

Sets up the gene patterns of the parents.



```
(*$S+*)
PROGRAM GENEPOOL;
(*$C Copyright 1980 JDEisenberg *)

USES TURTLEGRAPHICS,APPLESTUFF;

CONST
   GENE0=0;      GENE1=1;
   WIDTH=35;     HEIGHT=32;  (* AREA OF A FACE *)
   X0=0;         Y0=148;     (* BASE X,Y FOR FACES *)
   BEL=7;                    (* ASCII BELL CODE *)
   ESC = 27;                 (* CODE FOR ESCAPE KEY *)

TYPE
   SEXTYPE=(MALE,FEMALE);
   GENETYPE=(DOMINANT,RECESSIVE);
   CHARACTERISTIC=(FACE,HAIR);
   ZYGOTE=
      RECORD
        SEX:SEXTYPE;
        (* EACH TRAIT HAS TWO GENES *)
        TRAITS:ARRAY[FACE..HAIR,0..1] OF GENETYPE;
        SHOWSUP:ARRAY[FACE..HAIR] OF GENETYPE;
      END;

VAR
   DRAWCOLOR:SCREENCOLOR;
   COLOURS:
      ARRAY[FACE..HAIR,DOMINANT..RECESSIVE] OF SCREENCOLOR;
   KEY:CHAR;
   MOM,DAD:ZYGOTE;
   THEKIDS:ARRAY[0..23] OF ZYGOTE;
   STATS: ARRAY[FACE..HAIR] OF RECORD
             NHOMDOM,    (* NUMBER HOMOZYGOUS DOMINANT *)
             NHETDOM,    (* NUMBER HETEROZYGOUS DOMINANT *)
             NHOMREC:    (* NUMBER HETEROZYGOUS RECESSIVE *)
                INTEGER;
          END;
   RESTART,
   NOPADDLES:BOOLEAN;
   NKIDS,         (* SIZE OF THEKIDS ARRAY *)
   NMALE,         (* HOW MANY MALES *)
   NFEMALE,       (* HOW MANY FEMALE *)
   ROWSIZE,       (* # TO DISPLAY PER ROW *)
   COLOFFSET:     (* IN ORDER TO CENTRE FACES *)
      INTEGER;

PROCEDURE SKIPTO(X,Y:INTEGER);
BEGIN
   PENCOLOR(NONE); MOVETO(X,Y); PENCOLOR(DRAWCOLOR)
END;

PROCEDURE DRAWFACE(PERSON:ZYGOTE;ATROW,ATCOL:INTEGER);
VAR

   GENE:INTEGER;
   TRAIT:CHARACTERISTIC;
   XBASE,YBASE:INTEGER;     (* WHERE X,Y LEFT CORNER IS NOW *)
   INX,                     (* WHICH FILLER LINE TO DRAW *)
   DRAWLEN:INTEGER;         (* HOW LONG TO DRAW THE LINE *)
BEGIN
   (* DRAW THE FACE FIRST *)
   XBASE:=X0+ATCOL*WIDTH+5;
   YBASE:=Y0-(ATROW*HEIGHT)+22;
   DRAWCOLOR:=COLOURS[FACE,PERSON.SHOWSUP[FACE]];
   FOR INX:=0 TO 5 DO BEGIN
      SKIPTO(XBASE,YBASE-INX);
      MOVE(13)
   END;
   DRAWLEN:=13;
   FOR INX:=6 TO 9 DO BEGIN
      SKIPTO(XBASE+(13-DRAWLEN),YBASE-INX);
      MOVETO(XBASE+DRAWLEN,YBASE-INX);
      IF PERSON.SEX=FEMALE THEN DRAWLEN:=DRAWLEN-1
   END;
   (* SHOULDERS *)
   DRAWCOLOR:=VIOLET;
   IF PERSON.SEX=MALE THEN DRAWLEN:=24 ELSE DRAWLEN:=22;
   XBASE:=X0+ATCOL*WIDTH;
   SKIPTO(XBASE+10,YBASE-10); MOVE(5);
   FOR INX:=11 TO 12 DO BEGIN
      SKIPTO(XBASE+(24-DRAWLEN),YBASE-INX);
      MOVETO(XBASE+DRAWLEN,YBASE-INX);
      IF PERSON.SEX=FEMALE THEN DRAWLEN:=DRAWLEN+2
   END;
   (* NOW DRAW THE HAIR *)
   DRAWCOLOR:=COLOURS[HAIR,PERSON.SHOWSUP[HAIR]];
   XBASE:=X0+ATCOL*WIDTH+6; YBASE:=Y0-(ATROW*HEIGHT)+23;
   INX:=0;
   WHILE INX < 13 DO BEGIN
      SKIPTO(XBASE+INX,YBASE);
      IF PERSON.SEX=FEMALE THEN MOVETO(XBASE+6,YBASE+3)
      ELSE MOVETO(XBASE+INX,YBASE+3);
      INX:=INX+4
   END;
      (* NOW DRAW EYES/NOSE/MOUTH *)
      DRAWCOLOR:=BLACK2;
      SKIPTO(XBASE+3,YBASE-3); MOVETO(XBASE+3,YBASE-3);
      SKIPTO(XBASE+8,YBASE-3); MOVETO(XBASE+8,YBASE-3);
      SKIPTO(XBASE+6,YBASE-7); MOVETO(XBASE+6,YBASE-9);
   END;

   PROCEDURE DRAWGENES(PERSON:ZYGOTE;ATROW,ATCOL:INTEGER);
   VAR
      GENE:INTEGER;
      TRAIT:CHARACTERISTIC;
      XBASE,YBASE:INTEGER;     (* WHERE X,Y LEFT CORNER IS NOW *)
      INX:INTEGER;             (* WHICH FILLER LINE TO DRAW *)
   BEGIN


      (* COLOUR IN THE GENE SQUARES *)
      FOR TRAIT:=FACE TO HAIR DO BEGIN
         YBASE:=Y0-(ATROW*HEIGHT)+ORD(TRAIT)*5;
         XBASE:=X0+ATCOL*WIDTH;
         FOR GENE:=GENE0 TO GENE1 DO BEGIN
            DRAWCOLOR:=COLOURS[TRAIT,PERSON.TRAITS[TRAIT,GENE]];
            FOR INX:=0 TO 3 DO BEGIN
               SKIPTO(XBASE,YBASE+INX);
               MOVE(9)
            END;
            XBASE:=XBASE+14
         END (* FOR GENE *)
      END (* FOR TRAIT *)
   END;

   PROCEDURE DRAWME(PERSON:ZYGOTE;ATROW,ATCOL:INTEGER);
   BEGIN
      DRAWFACE(PERSON,ATROW,ATCOL);
      DRAWGENES(PERSON,ATROW,ATCOL)
   END;

   PROCEDURE AT(COL,ROW:INTEGER);
   BEGIN
      PENCOLOR(NONE);
      MOVETO(COL*7,184-ROW*8)
   END;

   PROCEDURE ERSLINE(ROW:INTEGER);
   BEGIN
      AT(0,ROW);
      WSTRING('
   END;

   PROCEDURE PSPACEBAR;
   BEGIN
      AT(3,23);
      WSTRING('Press spacebar to continue
      REPEAT
         READ(KEYBOARD,KEY)
      UNTIL (KEY=' ') OR (KEY=CHR(ESC));
      ERSLINE(23)
   END;

   PROCEDURE GENERATION;
   VAR
      INX:INTEGER;
      ASEX:SEXTYPE;
      TRAIT:CHARACTERISTIC;


   BEGIN
      (* DRAW THE PARENTS *)
      FILLSCREEN(BLACK);
      DRAWME(MOM,0,3); DRAWME(DAD,0,4);
      (* GENERATE THE  KIDS *)


   INX:=0;
   NKIDS:=23;
   NMALE:=1;    NFEMALE:=1; (* GUARANTEE ONE OF EACH *)
   FOR TRAIT:=FACE TO HAIR DO BEGIN
      STATS[TRAIT].NHOMDOM:=0;
      STATS[TRAIT].NHOMREC:=0;
      STATS[TRAIT].NHETDOM:=0;
   END;
   ROWSIZE:=8;
   COLOFFSET:=0;
   FOR INX:=0 TO 23 DO BEGIN
      WITH THEKIDS[INX] DO BEGIN
         IF INX=1 THEN SEX:=MALE
         ELSE IF INX=2 THEN SEX:=FEMALE
         ELSE IF (RANDOM MOD 2)=0 THEN BEGIN
```

```
   SEX:=MALE; NMALE:=NMALE + 1
END ELSE BEGIN
   SEX:=FEMALE; NFEMALE:=NFEMALE+1
END;


        FOR TRAIT := FACE TO HAIR DO BEGIN
           TRAITS[TRAIT,GENE0]:=MOM.TRAITS[TRAIT,RANDOM MOD 2];
           TRAITS[TRAIT,GENE1]:=DAD.TRAITS[TRAIT,RANDOM MOD 2];
           IF TRAITS[TRAIT,GENE0]=TRAITS[TRAIT,GENE1] THEN BEGIN
              SHOWSUP[TRAIT]:=TRAITS[TRAIT,GENE0];
              IF SHOWSUP[TRAIT]=RECESSIVE THEN
                 STATS[TRAIT].NHOMREC:=STATS[TRAIT].NHOMREC+1
              ELSE
                 STATS[TRAIT].NHOMDOM:=STATS[TRAIT].NHOMDOM+1
           END ELSE BEGIN
              SHOWSUP[TRAIT]:=DOMINANT;
              STATS[TRAIT].NHETDOM:=STATS[TRAIT].NHETDOM+1
           END
        END
     END; (* FOR *)
     (* AND DISPLAY THEM *)
     FOR INX:=0 TO NKIDS DO
        DRAWME(THEKIDS[INX],1+(INX DIV ROWSIZE),
          (INX MOD ROWSIZE)+COLOFFSET);
END;

PROCEDURE STATISTICS;
VAR KEY:CHAR;

   PROCEDURE SHOWZYGOTE(WHO:ZYGOTE;WHAT:CHARACTERISTIC);
   BEGIN
      IF WHAT=HAIR THEN
         WRITE('HAIR: ')
      ELSE
         WRITE('FACE: ');
      IF WHO.TRAITS[WHAT,GENE0]=WHO.TRAITS[WHAT,GENE1] THEN BEGIN
         WRITE('HOMOZYGOUS');
         IF WHO.TRAITS[WHAT,GENE0]=DOMINANT THEN
            WRITE(' DOMINANT')

         ELSE
            WRITE(' RECESSIVE')
      END ELSE BEGIN
         WRITE('HETEROZYGOUS DOMINANT');
      END
   END;


BEGIN
   (* WRITE INVISIBLY ON TEXT PAGE *)
   PAGE(OUTPUT);
   GOTOXY(1,1); WRITE('MOTHER');
   GOTOXY(10,2); SHOWZYGOTE(MOM,HAIR);
   GOTOXY(10,3); SHOWZYGOTE(MOM,FACE);
   GOTOXY(1,5); WRITE('FATHER');
   GOTOXY(10,6); SHOWZYGOTE(DAD,HAIR);
   GOTOXY(10,7); SHOWZYGOTE(DAD,FACE);
   GOTOXY(1,9); WRITE('OUT OF THESE 24 CHILDREN,');
   GOTOXY(6,10); WRITE(NMALE:2,' ARE MALE');
   GOTOXY(6,11); WRITE(NFEMALE:2,' ARE FEMALE');
   GOTOXY(1,13); WRITE('OFFSPRING''S HAIR');
   GOTOXY(10,14);
   WRITE('HOMOZYGOUS DOMINANT:  ',STATS[HAIR].NHOMDOM:3);
   GOTOXY(10,15);
   WRITE('HETEROZYGOUS DOMINANT:',STATS[HAIR].NHETDOM:3);
   GOTOXY(10,16);
   WRITE('HOMOZYGOUS RECESSIVE: ',STATS[HAIR].NHOMREC:3);
   GOTOXY(1,18); WRITE('OFFSPRING''S FACES');
   GOTOXY(10,19);
   WRITE('HOMOZYGOUS DOMINANT:  ',STATS[FACE].NHOMDOM:3);
   GOTOXY(10,20);
   WRITE('HETEROZYGOUS DOMINANT:',STATS[FACE].NHETDOM:3);
   GOTOXY(10,21);
   WRITE('HOMOZYGOUS RECESSIVE: ',STATS[FACE].NHOMREC:3);
   GOTOXY(2,23); WRITE('PRESS SPACEBAR TO CONTINUE');
   TEXTMODE; (* POOF-APPEARS LIKE MAGIC *)
   REPEAT
      READ(KEYBOARD,KEY)
   UNTIL KEY = ' ';
   GRAFMODE
END;

PROCEDURE LEGEND;
BEGIN
    FILLSCREEN(BLACK);
    AT(8,3); WSTRING('Mom ->');
    AT(25,3); WSTRING('<- Dad');
    DRAWME(MOM,0,3); DRAWME(DAD,0,4)
END;

PROCEDURE EXPLAIN;
VAR
   INX:INTEGER;
   EXPKIDS:ARRAY[0..3] OF ZYGOTE;

   PROCEDURE SETCOLOURS(DOMFACE,RECFACE,DOMHAIR,RECHAIR:SCREENCOLOR);
   BEGIN
      COLOURS[FACE,DOMINANT]:=DOMFACE;
      COLOURS[FACE,RECESSIVE]:=RECFACE;
      COLOURS[HAIR,DOMINANT]:=DOMHAIR;
      COLOURS[HAIR,RECESSIVE]:=RECHAIR
   END;
BEGIN
   FILLSCREEN(BLACK);

   (* SET UP ALL COMBINATIONS OF HAIR/FACE COLOURS *)
   FOR INX:=0 TO 3 DO BEGIN
      WITH EXPKIDS[INX] DO BEGIN
         IF (INX MOD 2)=0 THEN SEX:=MALE ELSE SEX:=FEMALE;
         CASE INX DIV 2 OF
            0: SHOWSUP[FACE]:=DOMINANT;
            1: SHOWSUP[FACE]:=RECESSIVE;
         END;
         CASE INX MOD 2 OF
            0: SHOWSUP[HAIR]:=DOMINANT;
            1: SHOWSUP[HAIR]:=RECESSIVE;
         END;
      END; (* WITH *)
      DRAWFACE(EXPKIDS[INX],0,INX+2);
   END;
   (* MAKE ONLY FACES SHOW UP *)
   SETCOLOURS(ORANGE,BLUE,NONE,NONE);
   EXPKIDS[0].SEX:=FEMALE;
   AT(1,6);  WSTRING('Above you see four Zorkons.');
   AT(1,8);  WSTRING('Zorkons may have green or white hair');
   AT(1,9);  WSTRING('and blue or orange faces.');
   AT(1,11); WSTRING('Orange face genes are');
   AT(1,12); WSTRING('dominant over blue face');
   AT(1,13); WSTRING('genes.');
   DRAWFACE(EXPKIDS[0],2,6); DRAWFACE(EXPKIDS[3],2,7);
   PSPACEBAR;
   (* MAKE ONLY HAIR SHOW UP *)
   SETCOLOURS(NONE,NONE,GREEN,WHITE1);
   DRAWFACE(EXPKIDS[0],2,6); DRAWFACE(EXPKIDS[3],2,7);
   (* RESTORE COLOURS TO NORMAL *)
   SETCOLOURS(ORANGE,BLUE,GREEN,WHITE1);
   AT(1,15); WSTRING('Green hair genes are dominant');
   AT(1,16); WSTRING('over white hair genes.');
   PSPACEBAR;
   AT(1,18); WSTRING('Male Zorkons have square');
   AT(1,19); WSTRING('faces and straight hair.');
   DRAWFACE(EXPKIDS[2],4,6);
   PSPACEBAR;
   AT(1,20); WSTRING('Female Zorkons have round');
   AT(1,21); WSTRING('faces and flat, piled-up hair.');
   DRAWFACE(EXPKIDS[3],4,7);
   PSPACEBAR
END;

PROCEDURE EXPLAIN2;


BEGIN
   LEGEND;
   AT(1,6);  WSTRING('This is a typical Zorkon couple,');
   AT(1,7);  WSTRING('Martha and Sam.');
   PSPACEBAR;
   AT(12,4); WSTRING('->');
   AT(12,5); WSTRING('->');
   AT(1,9);  WSTRING('The boxes beneath them tell you what');
   AT(1,10); WSTRING('genes they carry.');
   PSPACEBAR;
   AT(12,5); WSTRING('  ');
   AT(1,12); WSTRING('The top row shows their hair genes.');
   PSPACEBAR;
   AT(12,4); WSTRING('  ');
   AT(12,5); WSTRING('->');
   AT(1,13); WSTRING('The bottom row shows their face genes');
   PSPACEBAR;
   AT(12,5); WSTRING('  ');
   AT(1,15); WSTRING('Each Zorkon child wears two hair');
   AT(1,16); WSTRING('(and face) colour genes, one from');
   AT(1,17); WSTRING('from each parent.');
   PSPACEBAR;
END;

PROCEDURE EXPLAIN3;
VAR
   ONEKID:ZYGOTE;
   TRAIT:CHARACTERISTIC;
   GENE:INTEGER;
BEGIN
```

```
ONEKID.SEX:=MALE;
FOR TRAIT:=FACE TO HAIR DO BEGIN
   FOR GENE:=GENE0 TO GENE1 DO BEGIN
      ONEKID.TRAITS[TRAIT,GENE]:=DOMINANT;
   END;
   ONEKID.SHOWSUP[TRAIT]:=DOMINANT
END;



   FILLSCREEN(BLACK);
   DRAWME(ONEKID,0,3);
   AT(1,6);  WSTRING('If both genes for hair (or face) are');
   AT(1,8);  WSTRING('dominant, then the child is called');
   AT(1,10); WSTRING('HOMOZYGOUS DOMINANT for that trait.');
   PSPACEBAR;
   FOR TRAIT:=FACE TO HAIR DO BEGIN
      FOR GENE:=GENE0 TO GENE1 DO BEGIN
         ONEKID.TRAITS[TRAIT,GENE]:=RECESSIVE;
      END;
      ONEKID.SHOWSUP[TRAIT]:=RECESSIVE
   END;
   DRAWME(ONEKID,0,4);
   AT(1,12); WSTRING('If both genes are recessive, then the');
   AT(1,14); WSTRING('child is called HOMOZYGOUS RECESSIVE.');
   PSPACEBAR;
   FOR TRAIT:=FACE TO HAIR DO BEGIN

      ONEKID.TRAITS[TRAIT,GENE0]:=DOMINANT;
      ONEKID.SHOWSUP[TRAIT]:=DOMINANT
   END;
   DRAWME(ONEKID,0,5);
   AT(1,16); WSTRING('Finally, if the child gets one dominant');
   AT(1,18); WSTRING('and one recessive gene for a trait,');
   AT(1,20); WSTRING('the child is HETEROZYGOUS DOMINANT.');
   PSPACEBAR
END;

PROCEDURE EXPLAIN4;
BEGIN
   LEGEND;
   AT(1,6); WSTRING('You may select male and female genetic');
   AT(1,7); WSTRING('types, and a random sampling of their');
   AT(1,8); WSTRING('offspring will be produced.');
   AT(1,10);
   IF NOPADDLES THEN
      WSTRING('Use the arrow keys to select')
   ELSE
      WSTRING('Use paddle zero to select');
   AT(1,11); WSTRING('the genetic types.');
   PSPACEBAR
END;

PROCEDURE SPLASHPAGE;
VAR INX:INTEGER;
BEGIN
   GENERATION;
   AT(7,1);  WSTRING('Welcome to the Gene Pool');
   AT(2,20); WSTRING('A simulation of a genetics experiment');
   AT(5,22); WSTRING('Copyright 1980 by JDEisenberg');
   AT(1,23); WSTRING('Press spacebar to start, ? for help');
   REPEAT
      READ(KEYBOARD,KEY)
   UNTIL (KEY=' ') OR (KEY='?');
   IF KEY='?' THEN BEGIN
      EXPLAIN; EXPLAIN2; EXPLAIN3;
      EXPLAIN4; GENERATION
   END ELSE BEGIN
      ERSLINE(1); ERSLINE(20);
      ERSLINE(22); ERSLINE(23)
   END
END;

PROCEDURE INITPARENTS;
BEGIN
   (* SET UP MOTHER WITH ALL THE GENES *)
   WITH MOM DO BEGIN
      SEX:=FEMALE;
      TRAITS[HAIR,GENE0]:=DOMINANT;
      TRAITS[HAIR,GENE1]:=RECESSIVE;
      TRAITS[FACE,GENE0]:=DOMINANT;
      TRAITS[FACE,GENE1]:=RECESSIVE;

      SHOWSUP[HAIR]:=DOMINANT;
      SHOWSUP[FACE]:=DOMINANT
   END;


   (* SET UP FATHER SAME AS MOM, BUT MALE *)
   DAD:=MOM;
   DAD.SEX:=MALE
END;
```

```
PROCEDURE CHOOSETYPES;
VAR
   LASTPOS,POSITION:INTEGER;
   GOTCHA,HADERROR:BOOLEAN;

   PROCEDURE DRAWCURSOR;
   VAR ROW,COL,XBASE,YBASE:INTEGER;
   BEGIN
      ROW:=(POSITION DIV ROWSIZE)+1;
      COL:=(POSITION MOD ROWSIZE) + COLOFFSET;
      XBASE:=XO+COL*WIDTH+11;
      YBASE:=YO-(ROW*HEIGHT)+4;
      SKIPTO(XBASE-3,YBASE); MOVETO(XBASE+3,YBASE);
      SKIPTO(XBASE,YBASE-3); MOVETO(XBASE,YBASE+3)
   END;

   PROCEDURE GETPOSITION;
   BEGIN
      IF KEYPRESS THEN BEGIN
         READ(KEYBOARD,KEY);
         IF KEY=CHR(ESC) THEN EXIT(CHOOSETYPES);
         IF EOLN(KEYBOARD) THEN BEGIN
            RESTART:=TRUE;
            EXIT(CHOOSETYPES)
         END;
         IF KEY='S' THEN BEGIN
            STATISTICS
         END;
         IF NOPADDLES THEN BEGIN
            GOTCHA:=(KEY=' ');
            IF ORD(KEY)=8 THEN POSITION:=POSITION-1
            ELSE IF ORD(KEY)=21 THEN POSITION:=POSITION+1;
            IF POSITION < 0 THEN POSITION:=NKIDS;
            IF POSITION > NKIDS THEN POSITION:=0;
         END
      END;
      IF NOT NOPADDLES THEN BEGIN (* UNGRAMMATICAL BUT CORRECT *)
         POSITION:=PADDLE(0);

         (* MAKE SURE PADDLE REALLY IS MOVING *)
         IF ABS(POSITION-LASTPOS)>3 THEN LASTPOS:=POSITION;

         POSITION:=LASTPOS DIV (240 DIV NKIDS);
         IF POSITION > NKIDS THEN POSITION:=NKIDS;
         GOTCHA:=BUTTON(0)

      END
   END;

BEGIN
   LASTPOS:=0; POSITION:=0;
   HADERROR:=FALSE;
   AT(3,20);
   WSTRING('Press S for statistics');
   AT(3,21);
   WSTRING('Press RETURN to start again');
   AT(3,22); WSTRING('Press ESC to exit');
   REPEAT
      GOTCHA:=FALSE;
      AT(3,18); WSTRING('Choose the new male genetic type');
      AT(3,19);
      IF NOPADDLES THEN
         WSTRING('<-,-> to move; spacebar to choose')
      ELSE
         WSTRING('Press button to choose');
      DRAWCOLOR:=REVERSE;
      REPEAT
         GETPOSITION;
         DRAWCURSOR;
         DRAWCURSOR
      UNTIL GOTCHA;
      IF THEKIDS[POSITION].SEX <> MALE THEN BEGIN
         WRITE(CHR(BEL)); (* K'DING *)
         IF NOT HADERROR THEN BEGIN
            CHARTYPE(5);     (* REVERSE VIDEO *)
            AT(3,23); WSTRING('Males have square faces');
            CHARTYPE(10);    (* NORMAL VIDEO *)
            HADERROR:=TRUE
         END;
      END
      UNTIL THEKIDS[POSITION].SEX=MALE;
      DAD:=THEKIDS[POSITION];
      DRAWCURSOR;
      IF HADERROR THEN ERSLINE(23);
      HADERROR:=FALSE;
      REPEAT
         GOTCHA:=FALSE;
         AT(3,18); WSTRING('Choose the new female genetic type.');
         REPEAT
            GETPOSITION;
            DRAWCURSOR;
            DRAWCURSOR
```

```
    UNTIL GOTCHA;
    IF THEKIDS[POSITION].SEX <> FEMALE THEN BEGIN
       WRITE(CHR(BEL));  (* K'DING *)
       IF NOT HADERROR THEN BEGIN
          CHARTYPE(5);  (* REVERSE VIDEO *)
          AT(3,23); WSTRING('Females have rounded faces');
          CHARTYPE(10); (* NORMAL VIDEO *)
          HADERROR:=TRUE;
          END;
       END
    UNTIL THEKIDS[POSITION].SEX=FEMALE;
    MOM:=THEKIDS[POSITION];
    DRAWCURSOR
END;

PROCEDURE INITIALISE;
VAR POS0,POS1,DELAY:INTEGER; (* FOR READING PADDLES *)
BEGIN
    INITTURTLE;

    (* IF BOTH PADDLES READ 255, AND BOTH BUTTONS *)
    (* ARE PRESSED, THEN IT'S A SAFE BET THAT YOU *)
    (* DON'T HAVE PADDLES ATTACHED                *)
    POS0:=PADDLE(0);
    FOR DELAY:=1 TO 50 DO BEGIN END;
    POS1:=PADDLE(1);
    NOPADDLES:= ((POS0=255) AND (POS1=255)) AND
                ((BUTTON(0)) AND (BUTTON(1)));

    (* SET UP COLOURS FOR FACE/HAIR DISPLAY *)
    COLOURS[HAIR,DOMINANT]:=GREEN;
    COLOURS[HAIR,RECESSIVE]:=WHITE1;
    COLOURS[FACE,DOMINANT]:=ORANGE;
    COLOURS[FACE,RECESSIVE]:=BLUE;

    RESTART:=FALSE;
    INITPARENTS
END;

BEGIN
    RANDOMIZE;
    REPEAT
       INITIALISE;
       SPLASHPAGE;
       REPEAT
          CHOOSETYPES;
          IF (NOT RESTART) AND (KEY <> CHR(ESC)) THEN
             GENERATION
       UNTIL RESTART OR (KEY=CHR(ESC));
    UNTIL KEY=CHR(ESC);
    PAGE(OUTPUT);
    TEXTMODE
END.
```

PROCEDURE CHOOSETYPES;

This procedure lets you choose the new parents from the current generation. Of interest is the boolean variable (HADERROR, which tells if an error message has already been given or not (for choosing male instead of female or vice-versa).

PROCEDURE DRAWCURSOR;

Draws the blinking cursor while choosing parents. It calculates the appropriate ROW and COLumn on the screen from the POSITION (see GETPOSITION below).

PROCEDURE GETPOSITION;

Determines which of the offspring the cursor is pointing to, and returns its POSITION. Offspring are numbered left to right, top to bottom, starting with zero at the top left.

PROCEDURE INITIALISE;

Determines whether you have paddles attached or not (yes, you can fool this procedure — but why do it?); initializes display colors, and sets up genetic patterns of parents with a call to INITPARENTS.

Glossary

These terms are defined as applies to this article; for more complete definitions, I suggest that you go to your local library and read some books about genetics.

DOMINANT: having precedence over another; for example, brown eye genes are dominant over blue eye genes.

GENE: a hereditary unit that has a specific influence on a trait.

HETERO-: a prefix from the Greek, meaning "different."

HETEROZYGOUS: means a zygote that has different genes in it.

HETEROZYGOUS DOMINANT: a zygote with one dominant and one recessive gene in it. The dominant trait shows up in a person with this combination.

HOMO-: a prefix from the Greek, meaning "same."

HOMOZYGOUS: means a zygote that has both genes the same.

HOMOZYGOUS DOMINANT: a zygote with two dominant genes; the dominant trait shows up in a person with this combination.

HOMOZYGOUS RECESSIVE: a zygote with two recessive genes; the recessive trait shows up in a person with this combination.

RECESSIVE: opposite of dominant; thus blue eye genes are recessive.

TRAIT: a characteristic of a living being, such as hair color, etc.

ZYGOTE: a cell with two genes in it. Most zygotes appear in living beings, but they may also be found in crossword puzzles and Scrabble games (usually on triple word scores, and shortly thereafter on the floor). □

Wasps and The Impossible Elevator

Michael J. Orlove

The two programs described here demonstrate two interesting topics in genetics. In the program *Wasps*, the concepts of "altruism" genes and "selfishness" genes are introduced. An altruism gene is any gene which enables, allows, predisposes or causes an animal to give up its own reproduction, survival or parental care in order to augment the reproductive, parental caring, or survival-oriented activities of someone else.

This is not exactly the same thing as "altruism" in the original sense of the word; perhaps the word "beneficence," which means "having good effects on others," would be more appropriate. "Altruism" in the original sense is equivalent to "benevolence," which means "having the will or conscious intention to do good for others." We are, however, stuck with the historical precedent of the use of the word "altruism" in the biological literature.

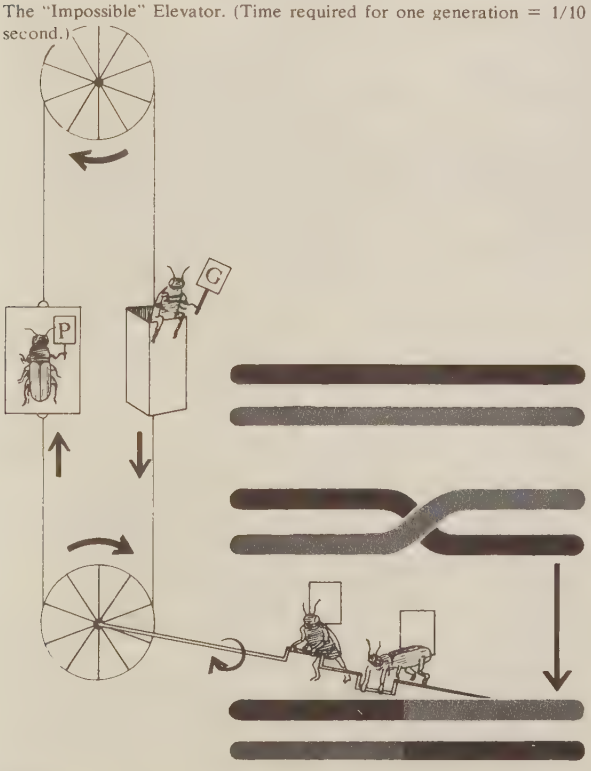
To read more about it, see Dawkins 1976 and Hamilton 1964. To read about altruism in polistes wasps, see West-Eberhard (1967, 1969), and a forthcoming article by Joan Strassmann (in press).

Michael Orlove, Dept. of Biology, Case Western Reserve University, Cleveland, OH 44106.

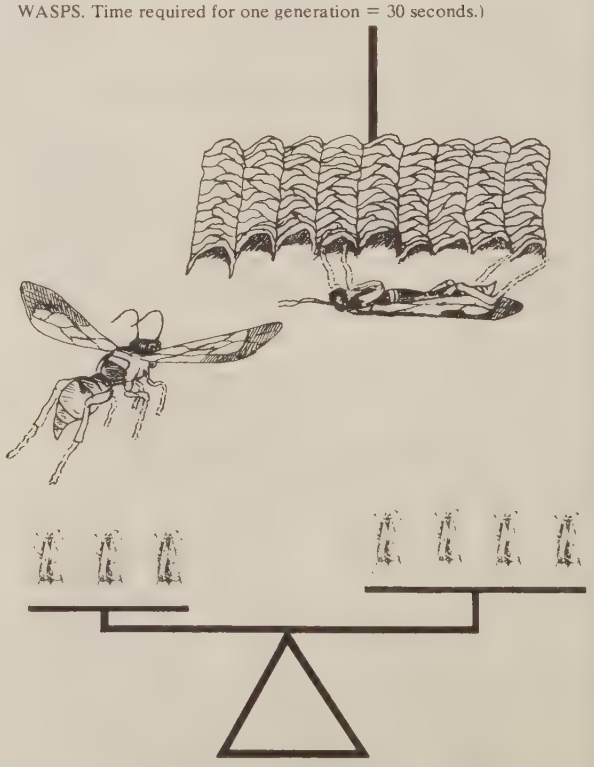
The Impossible Elevator

The Elevator Effect is a phenomenon in evolutionary genetics. It sounds as impossible as the perpetual motion machine, but it really works. The basic process of natural selection occurs when a gene increases in its relative abundance as compared to its alternatives; e.g. green eyes becoming commoner while blue eyes and brown eyes become rarer. An increase of this kind is called an increase in gene frequency. Such increases usually occur due to selection, i.e. when one gene makes the animal fitter in some way. Increases also occur when a useless gene is located next to a useful one. If a gene for purple eyelashes was located on the same chromosome next to a gene for immunity to measles, the purple gene would get a free ride. This is called hitchhiking. Both the rider and the gene providing the ride increase. The elevator effect occurs when the useless gene gets a free ride up (like an elevator car) and its alternative goes down in frequency (like the counterweight) and the gene providing the impetus (like

The "Impossible" Elevator. (Time required for one generation = 1/10 second.)



WASPS. Time required for one generation = 30 seconds.)



the motor) goes nowhere (stays at the same frequency).

The program simulates a population of beetles. In one position on the chromosome is a gene which makes a beetle spend its life rearing twenty nieces and nephews. The alternative gene makes the beetle raise ten of its own kids instead. Since nieces are half as expensive to raise, and also half as related to the rearer, both strategies are equally fit.

Located on the same chromosome is another position with the green-wing gene or the purple-wing gene. Initially, all chromosomes with niece-rearer genes have purple-wing genes and all chromosomes with offspring-rearer genes have green-wing genes. This relationship fades as time passes due to genes crossing over between chromosomes. Thus more and more chromosomes are found each generation with green and niece rearing or purple and kid rearing. Although no gene in this model is more useful than any other, the purple gene increases at the expense of the green. The niece-rearing and offspring-rearing genes stay at the same frequencies.

The proof of this effect is given in a paper in the *Journal of Theoretical Biology* (vol. 90 p. 81-100). To fully appreciate the program, it helps (but isn't necessary) to understand how genes cross over between chromosomes. They do *not* do it like people crossing the street, but by chromosomes exchanging whole segments with one another as in Figures 1 and 2. The places where chromosomes touch to be cut and spliced are called *chiasmata*. Figure 1 shows that an odd number of chiasmata between two chromosomal portions is successful. Figure 2 illustrates how an even number of chiasmata cancel each other out, rather like a double negative.

Thus if the number of units of chromosome length between two portions is 2 and the probability of a successful cross over per unit is 0.1, the net probability of a successful crossover is 0.18, not 0.2. The loss is due to cases where an even number of chiasmata exist.

Finally, I should give some space to the idea of 20 nieces or nephews being as precious as ten offspring. The pedigree diagram will elucidate. (See Diagram 1.)

Diagram 1.

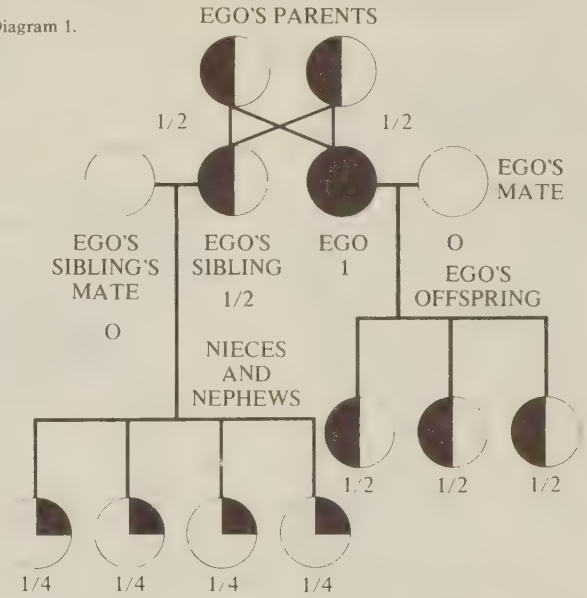


Figure 1.

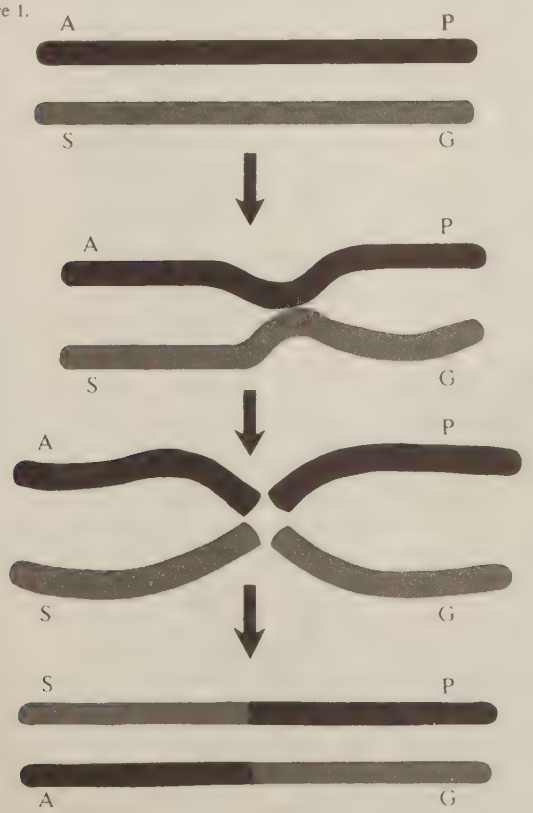
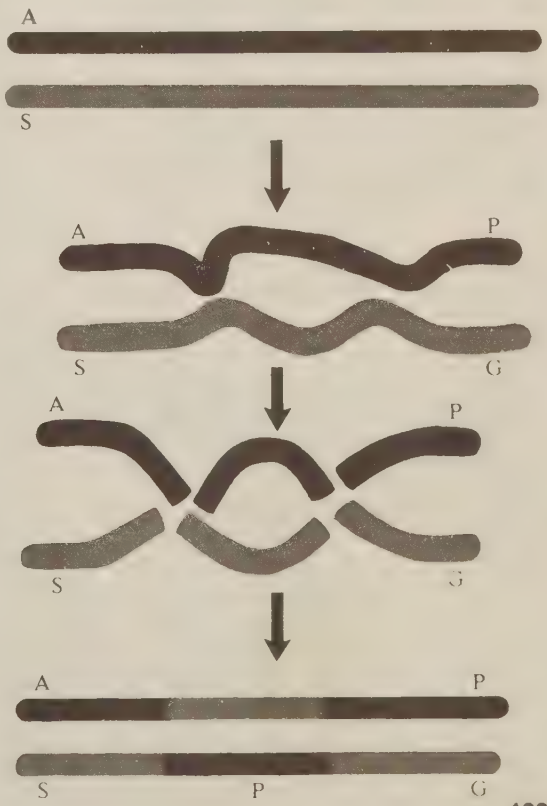


Figure 2.



The pie graphs and fractions indicate the degree to which Ego's genes can be found in each of Ego's relatives. According to modern interpretations of Darwin, animals will attempt to maximize their fitnesses (i.e. evolutionary success) by maximizing their number of descendants. According to Hamilton, this is not always so. Instead, an animal is said to maximize its fitness by maximizing the amount of shaded space (as in the pie graphs). This is how 20 nieces or nephews equals ten offspring.

In wasps, ants, bees, and their allies, males produce sperm asexually. As a result of this, sisters are halfway between ordinary sibs (0.5) and identical twins (1) and come out as 0.75. (See Diagram 2.)

The males are depicted as "half moons" because in this group of insects males have half the number of chromosomes that females do. This is why they reproduce asexually. This pedigree and Hamilton's theory explain many things such as why an animal will raise nieces and nephews instead of offspring, and why it will do so more readily if it is a bee, wasp, or ant. Also, this explains why in bees, wasps, and ants workers are females but males are lazy. In termites, workers and soldiers come in both sexes. The program *Wasps* will show you that, although Hamilton's theory predicts that four nieces or nephews (as opposed to the usual six in other animals) are worth three offspring. This actually only holds when there is no selection and altruism genes are extremely rare or extremely common or when altruism is half dominant. There are two ways to stop selection in the program. (See Diagram 3.)

When you enter "12345," the program recognizes this as a flag and replaces it with whatever value currently exists for number of offspring worth one niece or nephew. □

Diagram 2.

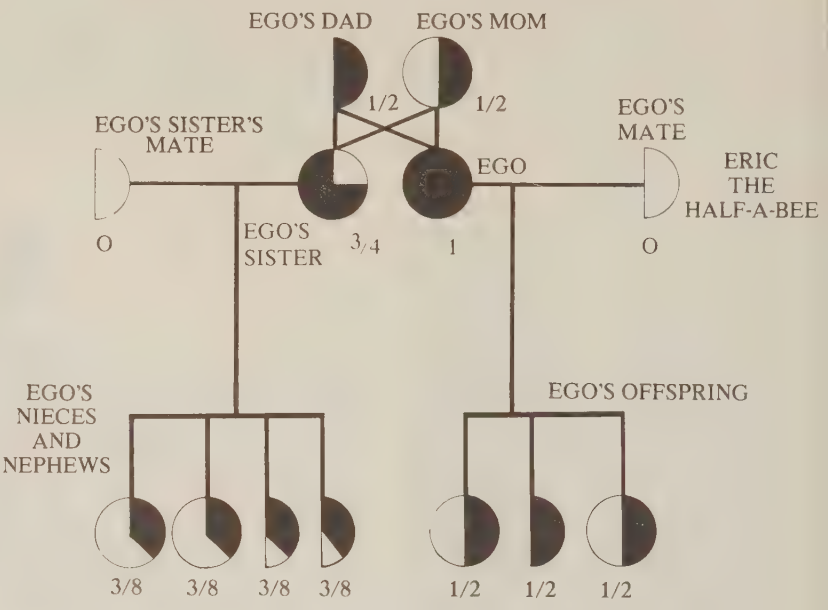


Diagram 3.

DATA ENTERED	method 1	method 2
Number of offspring given up by worker	0	12345
Number of offspring queen has anyway	ANY-THING	ANY-THING
Additional offspring queen has when helped by worker, i.e. nieces and nephews gained by worker	0	1

References

1) Dawkins, R., *The Selfish Gene*, Oxford University Press, Oxford, 1976.

2) Hamilton, W.D. "The Genetical Evolution of Social Behaviour I, II" *Journal of Theoretical Biology* 7, 1-16; p. 17-52.

3) Strassmann, J. "Kin Selection and Satellite Nests in Polistes exclamans" p. 45-58 in *Natural Selection and Social Behavior: Recent Results and Theory*, ed. R. D. Alexander and D. W. Tinkle, Chiron Press, New York, 1981.

4) West, M.J. (1967) *Science*, N.Y. 157, 1584.

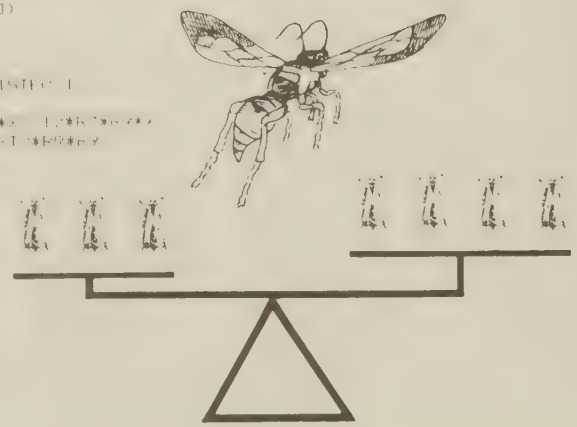
5) West Eberhard Misc. Publ. Mus. Zool., Univ., Mich. 140, 1, 1969.

Wasps

```
1 CLS:PRINT"W A S P S

BY MICHAEL ORLOVE

"
2 PRINT"THIS PROGRAM SIMULATES A POPULATION OF POLISTES WASPS.  EACH
FEMALE FORMS A COLONY OF TWO WITH HER SISTER.  A BRIEF SKIRMISH
OCCURS.  THE WINNER BECOMES THE QUEEN.  THE LOSER EITHER ACTS"
3 PRINT"AS A WORKER FOR HER SISTER OR FLIES OFF TO FOUND A NEST ALONE.
WHICH STRATEGY SHE CHOOSES DEPENDS ON WHETHER SHE HAS ALTRUISM
(NIECE RAISING) OR SELFISHNESS (DAUGHTER RAISING) GENES.
4 PRINT"THE GENE FREQUENCY REFERED TO IN THE DISPLAY IS THAT
FRACTION OF GENE THATS ARE ALTRUISM GENES."
9 INPUT"
<PRESS ENTER> ";Z:CLS
10 DIM N(8),X(8)
11 INPUT"SELECT DOMINANCE ON A SCALE OF 0 TO 1
0 BEING ALTRUISM DOMINANCE AND 1 BEING SELFISH DOMINANCE";S
14 INPUT"
NOW GIVE ME INITIAL GENE FREQUENCY (A NUMBER BETWEEN 0 AND 1)
";P
15 N(8)=P:N(7)=1-P
16 I0=1
20 O=P
40 FOR I=1TO6
60 N(I)=0
61 X(I)=0
65 NEXT I
70 X(7)=0:X(8)=0
71 INPUT"SEGREGATION RATIO (.5 IS NORMAL)";B2
72 INPUT"HOW MANY KIDS DOES A WORKER LOSE WHEN HELPING";W0
73 INPUT"HIW MANY KIDS DOES THE QUEEN MAKE BY HERSELF";Q0
74 INPUT"HOW MANY ADDITIONAL KIDS DOES THE QUEEN HAVE WHEN
THE WORKER HELPS HER";Q3
75 INPUT" TILL WHAT GENERATION SHOULD I RUN";I7
76 PRINT "GENE FREQ.","GENE FREQ.","OFFSPRING WORTH"," GENERATION"
77 PRINT"IN MALES","(POPULATION)","ONE NIECE"
80 FOR I3=I0 TO I7
90 GOSUB 260
100 W=W0:Q1=Q0:Q2=Q3
110 IF W<>12345 THEN 130
120 W=R
130 REM
140 A9=0
150 S9=0
160 FOR I=0TO1:FOR J=0TO1:FOR K=0TO1:FOR L=0TO1:FOR M=0TO1
170 I8=4*I+2*J+K+8*(L-I)*(1-J)*(1-K)
171 I9=4*A1+2*A2+A3+8*(1-A4)*(1-B2)*(1-B3)
180 X1=(W*(S*(I-J)↑2+I*J)+Q1+Q2*(1-(S*(J-K)↑2+J*K)))*N(I8)
185 B3=1-(B2*(I-J)↑2+I*J)
190 A9=A9+X1*B3:S9=S9+X1*(1-B3):X(I9)=X(I9)+X1*ABS((A1-B3)*(M-F)*(G-B3))
200 NEXT M,N,K,J,I:NEXT L:NEXT K:NEXT J:NEXT I
210 E=(2*P+O)/3:O=P:P=A9/(A9+S9)
220 PRINT O,P,R,I
230 D=0
240 FOR I=1TO8:D=D+X(I):NEXTI:FORI=1TO8:N(I)=X(I)/D:X(I)=0:NEXTI
250 NEXT I3
255 [illegible]
256 GOTO 72
260 N1=0,N2=0:[illegible]
270 FOR I =0TO1:FORJ=0TO1:FORK=0TO1
280 I8=4*I+2*J+K+8*(1-I)*(1-J)*(1-K)
290 W=W0*(1-J)+W1*J+W2*(1-J)*(1-K)+W3*J*K
300 F=B2*(I-J)↑2+I*J:T=1-F
310 [illegible]
320 D1=D1+W9*N(I8)*F:D2=D2+N(I8)*I
330 NEXTL:NEXTJ:NEXTI:R=N1*N2 N3*[illegible]
```

Impossible Elevator

```
1 DEFDBL A-H,O-Y
2 CLS:T1=1:T2=.2:T5=.5
5 DIM X(80)
10 PRINT CHR$(23);
20 PRINT@450,
30 INPUT"WHAT IS THE INITIAL FREQ. OF THE PURPLE GENE";P
35 CLS:PRINT@475,CHR$(23);
40 INPUT "HOW MANY UNITS OFCHROMOSOME LENGTH DO YOU WANT BETWEEN THE STRATEGY PO
SITION AND THE COLOR POSITION";N
41 L=10
42 IF N<100 THEN 45
43 L=80
45 CLS:PRINT@475,CHR$(23);
50 INPUT"WHAT IS THE PROBABILITY OF A SUCCESSFUL CROSSOVER PER UNIT";S
60 X8=N
70 R8=S
80 GOSUB 500
90 R=R7
95 CLS:PRINT@475,CHR$(23);
100 PRINT"THE NET PROBABILITY OF A SUCCESSFUL CROSSOVER IS ";:PRINT R
105 PRINT CHR$(23);
110 PRINT"THE NUMBER OF OWN KIDS WORTH ONE NIECE OR NEPHEW ( FROM THE VIEW POINT
 OF THE PURPLE GENE ) = ";:PRINT (T1/T2)/(T1-R)
120 P8=(T5*P*P+(T1+T5)*P)/(T1+P)
130 PRINT"THE FREQ. OF THE PURPLE GENE AT TIME=INFINITY IS "
140 PRINTP8
150 P1=P
155 INPUT "HOW MANY GENERATIONS SHOULD SIMULATION LAST";N
160 PRINT"FREQ. OF PURPLE", "GENERATION"
170 FOR I=1 TO N
180 PRINT P1;
181 PRINT I
190 P1=(P1-P8)*(T1-R)+P8
200 NEXT I
499 STOP
500 REM FIND NET CROSSOVER RATE HERE
540 FOR I=1TO
550 X8=X8/T2
560 NEXT I
570 FORJ=1TO
580 X8=X8*T2
590 I8=X8
600 X(J)=INT(I8)
610 X8=X8-X(J)
620 NEXT J
630 R7=0
640 R9=R8
650 FORK=1TO[illegible]
660 X7=X(K)
670 [illegible]
680 R9=1-R9*[illegible]
690 NEXT K
700 RETURN
```



Guess My Animal

Michael J. Orlove

This program plays "Guess My Animal" similar to the program in *Basic Computer Games*. Its internal method of storing questions and answers is somewhat different from the original version. The original stored the questions and possible answers in a linked list simulated with a character string array. This version, by Michael Orlove, stores the information in a tree or program-like form in a string array. The animal-guessing clues are stored as a set of nested IF . . . THEN . . . ELSE structures with the appropriate questions and animal name answers embedded.

At the start of the program run, the computer asks if you are writing the program. If you answer YES, the computer allows you to enter the animal-guessing tree expressed in its simple language. If you answer NO, then it will load a previously saved data tape. Once the game is running, it will recognize a set of commands as given below:

SAVE	Save animal-guessing information on data tape.
READ	Loads in previously saved data tape.
PRINT	Prints out the animal-guessing data tree.
MORE	Continues printing tree after stopping at end of screen.
FINISH	Stops printing tree as an alternate to MORE.
REPLACE	Replaces one string with another throughout the tree.

Note that the tape I/O routines are not implemented in this program but are indicated by remark statements and may be filled in by the user. In entering the program, please enter the words THEN, ELSE, and IF in lines 730, 770, and 810 as *lower case*. On the TRS-80 this must be done by holding down the shift key. The purpose of this subterfuge is to prevent the user from bombing the system by entering an animal named IF, THEN, or ELSE by converting the case. □

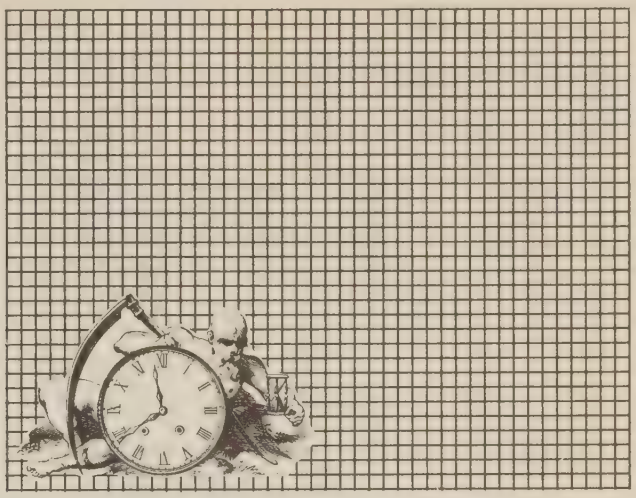
Mike Orlove, 2058 Powell Ave., Bronx, NY 10473.

```
ARE YOU WRITING THIS PROGRAM? YES
? IF
? DOES IT FLY?
? THEN
? PIGEON
? ELSE
? PIG
? OK
LET US PLAY 20 QUESTIONS, I GUESS AN ANIMAL AND
  YOU SAY YES OR NO.


DOES IT FLY?
? NO
IT MUST BE A    PIG              RIGHT?
? YES
LET US PLAY 20 QUESTIONS, I GUESS AN ANIMAL
  AND YOU SAY YES OR NO.


DOES IT FLY?
? YES
IT MUST BE A    PIGEON           RIGHT?
? NO
OK I LOSE THIS TIME, WHAT IS THE NAME OF
  YOUR ANIMAL? BEE
WHAT IS THE DIFFERENCE BETWEEN A PIGEON AND
  A BEE? THE BEE HAS SIX LEGS
TYPE IN AN APPROPRIATE QUESTION, WHICH
WHEN ANSWERED YES WILL INDICATE YOU ARE
THINKING OF THE BEE

WHEN YOU ARE FINISHED TYPE OK ON A LINE TO ITSELF.
? IS IT AN INSECT
? (DOES IT HAVE SIX LEGS)?
? OK
LET US PLAY 20 QUESTIONS, I GUESS AN ANIMAL
  AND YOU SAY YES OR NO.


DOES IT FLY?
? PRINT_
 IF
DOES IT FLY?
THEN
 IF
IS IT AN INSECT
(DOES IT HAVE SIX LEGS)?
THEN
BEE
ELSE
PIGEON
ELSE
PIG
OK


? _
```

```
 5 CLS
 9 CLEAR 3000
10 DIM V$(200)
12 X7=200
20 INPUT "ARE YOU WRITING THIS PROGRAM" ;A$
30 IF A$="NO"THEN 830
40 V$(1)="THEN":FORI=2TOX7
50 INPUT V$(I)
60 IFV$(I)="OK"THEN80
70 NEXTI
 80 PRINT"LET US PLAY 20 QUESTIONS,
    I GUESS AN ANIMAL AND YOU SAY YES OR NO. "
90 PRINT:PRINT:PRINT
100 T3=0:E3=0
110 I=1
120 I=I+1
130 IFV$(I)="IF"THEN170
140 IFV$(I-1)="THEN"THEN 320
150 IFV$(I-1)="ELSE"THEN 320
160 GOTO120
170 L=I+1:FORI=LTOX7
180 J=I:IFV$(I) ="THEN"THEN200
190 PRINTV$(I):NEXTI
200 INPUTA$:IFA$="YES"THEN120
201 IF A$="PRINT"THEN 690
202 IFA$="REPLACE"THEN900
203 REM HAVE IF A$=SAVE   REWIND TAPE THEN RECORD
    VECTOR V$
204 REM HAVE IF V$= LOAD   REWIND TAPE THE READ
    VECTOR V$
210 IFA$="SAVE"THEN 850
220 FORI=JTOX7
230 IFV$(I)="THEN"THEN 250
240 GOTO270
250 T3=T3+1
260 GOTO300
270 IFV$(I)="ELSE"THEN290
280 GOTO300
290 E3=E3+1
300 IFT3=E3THEN120
310 NEXTI
320 PRINT"IT MUST BE A ",V$(I)," RIGHT?"
330 INPUT A$:IFA$="YES"THEN 80
340 INPUT"OK I LOSE THIS TIME, WHAT IS THE NAME OF
    YOUR ANIMAL";A$
350 GOSUB 710
360 PRINT"WHAT IS THE DIFFERENCE BETWEEN A ";:PRINT
    V$(I);:PRINT" AND A ";:PRINT A$;
370 INPUT D$
380 PRINT"TYPE IN AN APPROPRIATE QUESTION, WHICH
    WHEN ANSWERED YES"
390 PRINT"WILL INDICATE YOU ARE THINKING OF THE ";
    :PRINT A$
400 PRINT"WHEN YOU ARE FINISHED TYPE OK ON A LINE
    TO ITSELF. "
 420 I$="IF":I9=I
 430 GOSUB620
 440 S$=A$
 449 L=I
 450 L=L+1
 460 INPUT A$
 470 GOSUB710
 480 I$=A$
 490 IFI$="OK"THEN530
 500 I9=L
 510 GOSUB620
 520 GOTO450
 530 L=L-1:I9=L+1
 540 I$="THEN"
 550 GOSUB620
 560 A$=S$
 570 I9=L+2:I$=A$
 580 GOSUB620
 590 I9=L+3:I$="ELSE"
 600 GOSUB620
 610 GOTO80
 620 M=I9-1
 621 M=M+1
 630 IFV$(M)="OK"THEN650
 640 GOTO621
 650 IF M=X7THEN 680
 660 FOR N=M+1TOI9STEP-1:V$(N)=V$(N-1):NEXTN
 670 V$(I9)=I$:RETURN
 680 REM USE TELETYPE
 690 X8=1
 691 FORH=X8TOX8+15:PRINT V$(H):NEXTH
 692 X8=X8+16
 693 INPUTX$
 694 IFX$<>"FINISH"THEN691
 695 GOTO80
 700 REM BACK TO SCREEN
 710 IFA$="THEN"THEN730
 720 GOTO750
 730 A$="THEN"
 740 RETURN
 750 IFA$="ELSE"THEN 770
 760 GOTO790
 770 A$="ELSE"
 780 RETURN
 790 IFA$="IF"THEN810
 800 RETURN
 810 A$="IF"
 820 RETURN
 830 REM READ V$ FROM TAPE
 840 GOTO80
 850 REM BACKSPACE 1
 860 REM SAVE V$ ON TAPE
 870 GOTO80
 900 INPUT A$,B$
 910 I=0
 920 I=I+1
 930 IFV$(I)="OK"THEN 80
 940 IFA$<>V$(I)THEN920
 950 V$(I)=B$
 960 GOTO80
```

Lines 730, 770 and 810: THEN, ELSE and IF should be entered as lower case in the TRS-80 by holding down the shift key while typing them!

Game of Life



Generally, a mathematical model is a representation of some real-life process, expressed in mathematical form (such as a set of related equations) or in algorithmic form (such as a computer program). Usually the model is by necessity a simplification of the actual process, since real-life processes tend to be highly complex. One advantage of embodying the model as a computer program is that we can run the program and thus simulate the process being modelled. By varying certain features of the program, we can learn something about the relationships between the components and the overall structure of the process. In addition, if the output does not sufficiently coincide with observed reality, the model can be revised and improved.

It is also possible to model a purely abstract process. We don't often see this done. After all, if someone asked you to describe some abstract process, what would you say? However, many games start out as purely abstract processes. For example, tic-tac-toe or checkers are abstract from the point of view that they represent no real-life process. Occasionally, it turns out that an abstract process represents a real-life process either by accident or design. The model described below is one such abstract game that in some ways actually represents life itself.

With special acknowledgement to Paul Shapiro of Newton High School in Newton, Massachusetts.

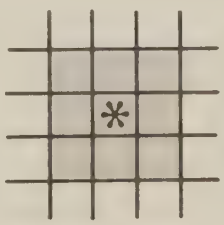
The Game of Life

The Game of "Life" was devised by John Conway, a mathematician at the University of Cambridge, and made popular by a series of articles written by Martin Gardner in past issues of *Scientific American*. Ever since the first article appeared in October 1970, hundreds of mathematicians throughout the world have become fascinated with the model and have been exploring its properties.

The game consists of following the successive generations of a particular imaginary type of cellular life-form. The life processes of these cells are represented by the following mathematical model, which captures several properties common to all life-forms.

(1) *World* — Cells live on an infinite two-dimensional plane of squares (like an infinite checker-board except that all squares are identical).

(2) *Neighborhood* — *Each* square has eight "neighbor" squares. In the diagram below, the *neighbor* squares for the square with the asterisk (*) have been colored in.



(3) *Survival* — A cell (always represented by a *) which is living in generation n, will remain living in generation n+1 if and only if it has exactly 2 or 3 living neighbors in generation n.

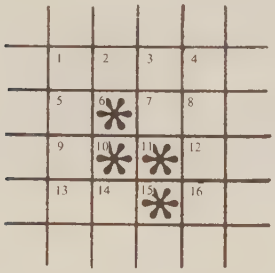
(4) *Death* — However, in all other cases the cell dies. Specifically: If it has 0 or 1 neighbors it dies from isolation. If it has 4, 5, 6, 7 or 8 neighbors, it dies from overpopulation.

(5) *Birth* — If a square is empty during generation n, a living cell will be born into that square during generation n+1 if and only if that square had *exactly 3* (no more, no less) living neighbors during generation n.

The only trick is to remember that all survivals, deaths, and births occur *simultaneously*, and so the simplest way to keep the "bookkeeping" straight is to have two separate copies of the world — one for the old generation and one for the new one you are forming. For each square in the old world, decide what its state will be next time, and mark this down *in the corresponding square* in the new world.

The game is played simply by picking some initial starting pattern and watching the development of some very interesting, and often beautiful, patterns of symmetry. However, the player must be extremely careful because mistakes are easy to make.

As an example, we will trace three generations of the following initial pattern (we have numbered some rows and columns for reference purposes only):



Following the rules of our model:

No births will occur in squares 1, 2, 3, 4 or 5 because none has *three* living neighbors.

The cell in square 6 will survive because it has two living neighbors (10 and 11).

A birth occurs in square 7 because there are three living neighbors (6, 10 and 11).

No birth occurs in squares 8 or 9.

The cell in square 10 survives because it has three living neighbors (6, 11 and 15).

The cell in square 11 survives also because it has three living neighbors (6, 10 and 15).

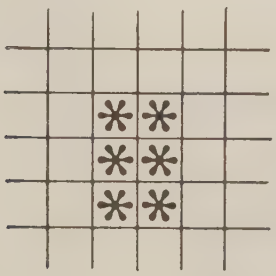
No birth occurs in squares 12 or 13.

A birth occurs in square 14 because there are three living neighbors (10, 11 and 15).

The cell in square 15 survives because it has two living neighbors (10 and 11).

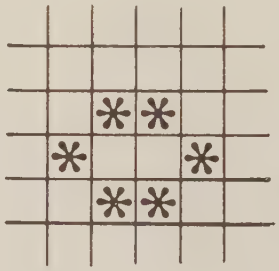
No birth occurs in square 16 because it only has two living neighbors.

During this process, we have been filling in a picture of G1, and the end result is:



Exercise 1

Using pencil and paper, carefully compute G2, the next generation for this same society of cells. If you do it correctly, you will find that G2 is the pattern which Conway calls the "beehive":

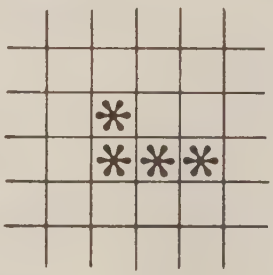


Exercise 2

Now compute G3. If you are again careful, you will discover that G3 is identical to G2. Why does Conway call the beehive a "still-life"? If you are not sure, think about G4, G5, G6 . . .

Exercise 3

Using pencil and paper, compute G0, G1, G2 and G3 for the initial pattern below:



If you do it correctly, G3 should look familiar to you.

Project 1

By now you've no doubt noticed that with pencil and paper, this game is an extremely slow process, and mistakes are all too common. If we ever hope to look at more than a few patterns, we're going to have to turn to the computer for help.

Write a computer program which simulates "Life" for *any* given initial

pattern, and which has the following features:

(1) Allow for as large a world size as your particular computer facility will permit (obviously an infinite plane is not possible in a finite memory). You will probably want to use array structures with two subscripts (row and column).

(2) Whatever world-size you are limited to, make sure your program doesn't try to allow births outside your world, even though properly these *would* occur on an infinite plane.

(3) Make sure your algorithm allows all survivals, deaths, and births during a given generation to occur *simultaneously*, as discussed above.

(4) Allow the user to input the initial pattern in a convenient format, such as pairs of (row, column) coordinates.

(5) Make your program efficient and your output as close to the format of the pictures above as possible.

Once your algorithm is designed, and your program is written, debug your program by running it on the following initial GO pattern, and carefully check your output vs. the results below:

Warning!

Depending upon the world-size you are limited to, certain "large" patterns may grow differently than they would on an infinite plane.

If the society of cells above, however, fits inside your world-size, you will notice an interesting cyclic pattern beginning at GO, which Conway calls "traffic lights."

Project 2

When your program is thoroughly debugged and operational, or using the existing LIFE program, the real fun comes in thinking up initial patterns and watching them grow. Interesting situations to watch for are:

(1) Other "still-life" societies (like the "bee-hive")

(2) Other "cyclic" societies (like the "traffic lights")

(3) A society which lives for an extended period of time without dying, becoming still, or cycling

Bring your findings to class for comparison and discussion.

Optional Project 3

Find copies of the October 1970 and/or February 1971 issues of *Scientific American* and read Gardner's articles on "Life." You may want to run your program on some of the societies he describes, such as: diagonal chains, the R pentomino, the Latin cross, the cheshire cat, and many others. Your school library may subscribe to the magazine. If not, the public library near you will certainly have it available.

Optional Project 4

Try to think up changes in the model (and your computer program) which will drastically alter the life patterns of the cells, i.e. by modifying the rules for birth or death or both. Based upon your experience so far, try to come up with sets of rules which will lead to more populous societies, or more sparse societies, or societies which are less symmetric than those of "Life", etc. The range of possibilities is very large. Bring your findings to class for comparison and discussion.

Optional Project 5

Make some major modifications in your computer program to make it more *general*, by allowing the user to specify the particular model he wants to investigate. For example, you might have your program begin by posing the following questions to the user:

How many neighbors for survival?

How many neighbors for birth?

Then, if the user answered 2, 3 for the first question and 3 for the second, your program would follow the rules of "Life." But if he gave other answers, the program would simulate for him some other model he wants to investigate.

Optional Project 6

Is there any way you can streamline your program so that you can enlarge the size of the world it currently handles?

Optional Project 7

How might you alter the general concept of "neighborhood" so that entirely different models could be tested? How would your computer program have to be changed in order to simulate these new models.

Optional Project 8

It is possible to place two different types of life forms in the world and let them interact with each other subject to the standard rules for LIFE. A competitive LIFE game, LIFE-2, has been devised which allows two players to put 3 life forms on the board per move and try to eliminate the opposing player. Class members may play this game as individuals or as teams. Are there any playin strategies which can be rigorously stated? Can two good players go on forever with no winner?

Life

The Game of Life was originally described in *Scientific American*, October 1970, in an article by Martin Gardner. The game itself was originated by John Conway of Gonville and Caius College, University of Cambridge, England.

In the "manual" game, organisms exist in the form of counters (chips or checkers) on a large checkerboard and die or reproduce according to some simple genetic rules. Conway's criteria for choosing his genetic laws were carefully delineated as follows:

1. There should be no initial pattern for which there is a simple proof that the population can grow without limit.
2. There should be initial patterns that apparently do grow without limit.
3. There should be simple initial patterns that grow and change for a considerable period of time before coming to an end in three possible ways: fading away completely (from overcrowding or from becoming too sparse), settling into a stable configuration that remains unchanged thereafter, or entering an oscillating phase in which they repeat an endless cycle of two or more periods.

In brief, the rules should be such as to make the behavior of the population relatively unpredictable. Conway's genetic laws are delightfully simple. First note that each cell of the checkerboard (assumed to be an infinite plane) has eight neighboring cells, four adjacent orthogonally, four adjacent diagonally. The rules are:

1. Survivals. Every counter with two or three neighboring counters survives for the next generation.
2. Deaths. Each counter with four or more neighbors dies (is removed) from overpopulation. Every counter with one neighbor or none dies from isolation.
3. Births. Each empty cell adjacent to exactly three neighbors — no more, no fewer — is a birth cell. A counter is placed on it at the next move.

It is important to understand that all births and deaths occur simultaneously. Together they constitute a single generation or, as we shall call it, a "move" in the complete "life history" of the initial configuration.

You will find the population constantly undergoing unusual, sometimes beautiful and always unexpected change. In a few cases the society eventually dies out (all counters vanishing), although this may not happen until after a great many generations. Most starting patterns either reach stable figures — Conway calls them "still lifes" — that cannot change or patterns that oscillate forever. Patterns with no initial symmetry tend to become symmetrical. Once this happens the symmetry cannot be lost, although it may increase in richness.

Conway used a DEC PDP-7 with a graphic display to observe long-lived populations. You'll probably find this more enjoyable to watch on a CRT than a hard-copy terminal.

Since MITS 8K BASIC does not have LINE INPUT, to enter leading blanks in the pattern, type a "." at the start of the line. This will be converted to a space by BASIC, but it permits you to type leading spaces. Typing DONE indicates that you are finished entering the pattern. See sample run.

Clark Baker of Project DELTA originally wrote this version of LIFE which was further modified by Steve North of Creative Computing.

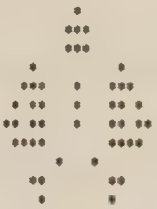
```
                  LIFE
 CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

ENTER YOUR PATTERN:
? .  ***
? . *  *
? .*   *
? DONE

GENERATION: 0                POPULATION: 7

 ***
*  *
*  *

GENERATION: 1                POPULATION: 8

 *
***
** **

GENERATION: 2                POPULATION: 9

 ***
*   *
** **

GENERATION: 3                POPULATION: 10

 *
***
*   *
** **

GENERATION: 4                POPULATION: 12

 ***
***
*   *
** **
```

GENERATION: 13 POPULATION: 30

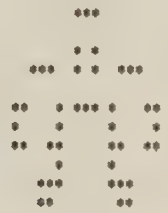
GENERATION: 14 POPULATION: 39

GENERATION: 15 POPULATION: 37

GENERATION: 16 POPULATION: 48

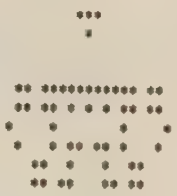


GENERATION: 17 POPULATION: 46

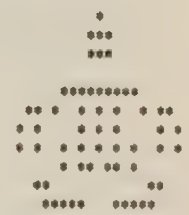


GENERATION: 18 POPULATION: 54

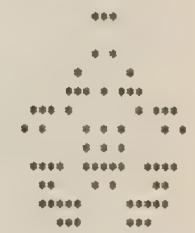
GENERATION: 19 POPULATION: 56



GENERATION: 20 POPULATION: 61



GENERATION: 21 POPULATION: 67



```
  2 PRINT TAB(34);"LIFE"
  4 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
  6 PRINT: PRINT: PRINT
  8 PRINT "ENTER YOUR PATTERN:"
  9 X1=1: Y1=1: X2=24: Y2=70
 10 DIM A(24,70),B$(24)
 20 C=1
 30 INPUT B$(C)
 40 IF B$(C)="DONE" THEN B$(C)="": GOTO 80
 50 IF LEFT$(B$(C),1)="." THEN B$(C)=" "+RIGHT$(B$(C),LEN(B$(C))-1)
 60 C=C+1
 70 GOTO 30
 80 C=C-1: L=0
 90 FOR X=1 TO C-1
100 IF LEN(B$(X))>L THEN L=LEN(B$(X))
110 NEXT X
120 X1=11-C/2
130 Y1=33-L/2
140 FOR X=1 TO C
150 FOR Y=1 TO LEN(B$(X))
160 IF MID$(B$(X),Y,1)<>" " THEN A(X1+X,Y1+Y)=1:P=P+1
170 NEXT Y
180 NEXT X
200 PRINT:PRINT:PRINT
210 PRINT "GENERATION:";G,"POPULATION:";P;: IF I9 THEN PRINT "INVALID";
215 X3=24: Y3=70: X4=1: Y4=1: P=0
220 G=G+1
225 FOR X=1 TO X1-1: PRINT: NEXT X
230 FOR X=X1 TO X2
240 PRINT
250 FOR Y=Y1 TO Y2
253 IF A(X,Y)=2 THEN A(X,Y)=0: GOTO 270
256 IF A(X,Y)=3 THEN A(X,Y)=1: GOTO 261
260 IF A(X,Y)<>1 THEN 270
261 PRINT TAB(Y);"*";
262 IF X<X3 THEN X3=X
264 IF X>X4 THEN X4=X
266 IF Y<Y3 THEN Y3=Y
268 IF Y>Y4 THEN Y4=Y
270 NEXT Y
280 NEXT X
295 FOR X=X2+1 TO 24: PRINT: NEXT X
299 X1=X3: X2=X4: Y1=Y3: Y2=Y4
301 IF X1<3 THEN X1=3: I9=-1
303 IF X2>22 THEN X2=22: I9=-1
305 IF Y1<3 THEN Y1=3: I9=-1
307 IF Y2>68 THEN Y2=68: I9=-1
309 P=0
500 FOR X=X1-1 TO X2+1
510 FOR Y=Y1-1 TO Y2+1
520 C=0
530 FOR I=X-1 TO X+1
540 FOR J=Y-1 TO Y+1
550 IF A(I,J)=1 OR A(I,J)=2 THEN C=C+1
560 NEXT J
570 NEXT I
580 IF A(X,Y)=0 THEN 610
590 IF C<3 OR C>4 THEN A(X,Y)=2: GOTO 600
595 P=P+1
600 GOTO 620
610 IF C=3 THEN A(X,Y)=3: P=P+1
620 NEXT Y
630 NEXT X
635 X1=X1-1: Y1=Y1-1: X2=X2+1: Y2=Y2+1
640 GOTO 210
650 END
```

A Faster Life

Arthur L. Thomas

Creative Computing's *Basic Computer Games* includes Baker and North's ingenious program for Conway's population dynamics simulation, Life. Unfortunately, on some machines this program is a bit sluggish. For instance, on my Wang PCS II it takes about 15½ minutes for it to generate the first 21 tableaux of Baker and North's sample run.

Much of this time is spent repeatedly executing program steps 530-570. The simulation uses a matrix, and the program identifies its "active" portion — a rectangle within which populations can change during the short run. For each cell in that rectangle, steps 530-570 obtain population information about both it and its eight neighboring cells. Thus, each cell that lies in the rectangle's interior will be investigated 9 times per tableau. During the first 21 tableaux, there are 25,992 such investigations!

The remedy is to change the way that one represents populations. The program FASTLIFE uses the same version of Basic, and makes the same assumptions, as the Baker-North program. But each cell of its primary (Z) matrix reflects both whether it and whether its neighbors are occupied. This allows the program to skip over all cells except those in which a birth or death is about to occur. FASTLIFE also saves a little more time by not doing the testing for illegal values performed by lines 301-307 of the Baker-North program. (Instead, an illegal value merely terminates the run.)

On the PCS II, FASTLIFE generates the first 21 tableaux in 7¼ minutes — more than twice as rapidly as the Baker-North program. It is especially efficient whenever populations are loosely packed, as when the center of the matrix is empty, or the population includes small, isolated stable groups. As an extreme example, for an 18 by 18 rectangle with a square of four occupied cells in each corner, the Baker-North program requires 101 seconds per iteration, while FASTLIFE needs only 21½.

Finally, if your Basic includes a few elementary matrix operations, you can make FASTLIFE run even faster. Interested readers are invited to request from me a PCS II program that generates the first 21 tableaux in less than five minutes, and the 18 by 18 rectangle in 13½ seconds.

Arthur L. Thomas, Rice University, P.O. Box 1892, Houston, TX 77001.

```
10 REM "FASTLIFE"
20 PRINT TAB(26); "J.H. CONWAY'S 'LIFE'": PRINT
30 REM PROGRAM BY ARTHUR L. THOMAS, THE UNIVERSITY OF KANSAS, THE
40 REM             SCHOOL OF BUSINESS, LAWRENCE, KANSAS  66045
50 PRINT TAB(6); "BEGIN EACH INPUT LINE WITH A DOT, THEN ASTERISKS AND
   SPACES."
60 PRINT TAB(10); "TO END INPUT, RESPOND TO THE PROMPT WITH / (RETURN)."
   :PRINT :PRINT
70 DIM Y(26,26), Z(26,26), A$(21)21
80 L = L+1
90 INPUT A$(L)
100 IF LEFT$(A$(L),1) = "/" THEN 150
110 IF W >= LEN(A$(L))-1 THEN 130
120 W = LEN(A$(L))-1
130 IF L = 20 THEN 150
140 GOTO 80
150 L = L-1: L1 = INT((26-L)/2)+1: W1 = INT(((26-W)/2)+1.5)
160 L9 = L+L1-1: W9 = W+W1-1
170 FOR I = L1 to L9
180 FOR J = W1 TO W9
190 IF MID$(A$(I-L1+1),(J-W1+2),1) = " " THEN 260
200 Z(I,J) = Z(I,J)+10
210 FOR I1 = -1 TO 1
220 FOR J1 = -1 TO 1
230 Z(I+I1,J+J1) = Z(I+I1,J+J1)+1
240 NEXT J1
250 NEXT I1
260 NEXT J
270 NEXT I
280 L2 = L1: L8 = L9: W2 = W1: W8 = W9: L1,L9 = INT((L1+L9)/2): W1,W9 =
    INT((W1+W9)/2): Q = 0
290 FOR I = 1 TO L2-2 :PRINT :NEXT I
300 FOR I = L2-1 TO L8+1
310 FOR J = W2-1 TO W8+1
320 IF Z(I,J) > 9 THEN 480
330 IF Z(I,J) <> 3 THEN 560
340 Y(I,J) = Y(I,J)+10
350 FOR I1 = -1 TO 1
360 FOR J1 = -1 TO 1
370 Y(I+I1,J+J1) = Y(I+I1,J+J1)+1
380 NEXT J1
390 NEXT I1
400 IF L1 < I THEN 420
410 L1 = I
420 IF L9 > I THEN 440
430 L9 = I
440 IF W1 < J THEN 460
450 W1 = J
460 IF W9 > J THEN 560
470 W9 = J: GOTO 560
480 Q = Q+1: PRINT TAB(J+22); "*";
490 IF ABS(Z(I,J)-13.5) = .5 THEN 400
500 Y(I,J) = Y(I,J)-10
510 FOR I1 = -1 TO 1
520 FOR J1 = -1 TO 1
530 Y(I+I1,J+J1) = Y(I+I1,J+J1)-1
540 NEXT J1
550 NEXT I1
560 NEXT J
570 PRINT
580 NEXT I
590 FOR I = L8+2 TO 22 :PRINT :NEXT I
600 PRINT TAB(20); "GENERATION"; G; TAB(39); "POPULATION"; Q; CHR$(07);
610 FOR I = L2-2 TO L8+2
620 FOR J = W2-2 TO W8+2
630 IF Y(I,J) = 0 THEN 660
640 Z(I,J) = Z(I,J)+Y(I,J)
650 Y(I,J) = 0
660 NEXT J
670 NEXT I
680 G = G+1
690 GOTO 280
```

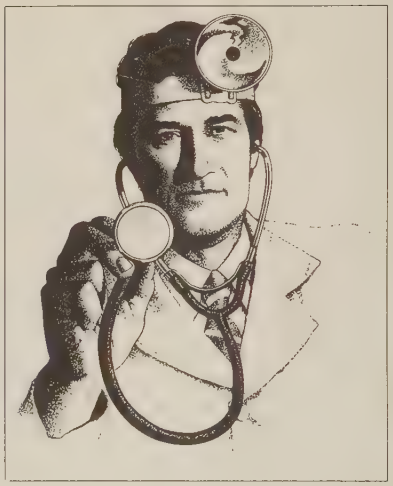
"Looks like it might be a nice day tomorrow!"

Chapter Three

Health and Psychology

Medical Simulations

Dr. Khurheed Ahmed and Dr. George Sweeney



You are traveling in a jet plane at 30,000 feet; suddenly the cabin blows open and atmospheric pressure falls — what happens to your breathing? How does your blood retain enough oxygen? How does the body respond? How long can you survive? In the past, the only way to answer these questions was to perform experiments on willing volunteers, or use animals for experimentation, or simply rely on the facts stated in books. However, with the development of computer simulation techniques, it is now possible to duplicate these experiments without any pain or danger to living beings. In fact, using these tools, medical students in many schools are performing experiments which would be unethical or technically impossible in real life. Such simulations can be very instructive for understanding human physiology and to study how our body responds in states of health and in disease.

Simulation techniques have been used in many other fields, such as space technology, engineering problems, economics, ecology, etc. For example, pilots are routinely trained to fly a jet plane in a flight simulator. However, the application of this concept to medicine and physiology is relatively new. This is partly because physiological systems are very difficult to describe mathematically. We must first determine the laws governing a system and the intricate relationships between all the varying factors, before we can program them into a computer — this is a formidable task. In practice, we build a mathematical 'model' of the system by making some assumptions. We then gradually refine this model and test its predictions against as many known facts as possible until we have achieved a reasonably realistic representation. This is the approach we have taken in developing a series of simulation models of human physiology.

In 1970, Dr. C. J. Dickinson, who came to McMaster University from London as the R. S. McLaughlin visiting professor, decided to attempt a computer simulation of the human heart and circulation of blood in the body. The initial attempts were encouraging and, after some program aborts and the usual debugging steps, a successful program emerged. This abstract creature was named MACMAN. He was soon tamed to allow some basic experiments dealing with loss of blood, heart-attack, hardening of the arteries and other abnormalities. He was also taught enough language to record some of his more serious symptoms as verbal comments. Graphic and numeric data

```
                ----- NEW (INTACT) SUBJECT -----
 XXXX SYSTOLIC AND DIASTOLIC PRESSURES-MM HG '*' HEART RATE-BEATS/MIN

 TIME 0  20  40  60  80  100 120 140 160 180 200 220 240 260 280 300
 SECS .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
  -28              *XXXXXXXXX
  -24              *XXXXXXXXX
  -20              *XXXXXXXXX
  -16              *XXXXXXXXXX
  -12              *XXXXXXXXXX
   -8              * XXXXXXXXXX
   -4              *XXXXXXXXXX
    0              * XXXXXXXXXX

 B.P. SYSTOLIC=121  DIASTOLIC= 75  MEAN=  90.7
 RIGHT ATRIAL PRESS.= 1.8 MM HG  MEAN CAPILL.PRESS.=    12.6 MM HG
 CARDIAC OUTPUT=  4.9 L/MIN, STROKE VOL.=  68.3 ML, HEART RATE=  72.
 ART.RESIST.=  15.9   VENOUS RESIST.=  2.2 MM HG/L/MIN
 CARDIAC CONTRACILITY=  1.3 L/MIN/MM HG FILLING PRESS.
 VALUES FOR THIS RUN WERE...

 1..ART. RESISTANCE=  100.0 % OF NORMAL
 2..VEN. RESISTANCE=  100.0 % OF NORMAL
 3..CARDIAC CONTRACTILITY=  100.0 % OF NORMAL
 4..MEAN INTRATHORACIC PRES.=  -2.0 MM HG
 5..BLOOD VOLUME=  5000.0 ML
```

Figure A **This is MACMAN in a normal, healthy state. His heart rate is 72 beats/minute, blood pressure is 121/75 mmHg, and he has 5 litres of blood in his body.**

Dr. Khursheed Ahmed / Dr. George Sweeney, McMaster University Medical School, Hamilton, Ontario, Canada L85 4J9.

were a natural output from his equations.

A typical encounter with MACMAN is shown in Figures A to D. Figure A shows MACMAN's heart beating normally at 72 beats per minute (shown as symbol*), and maintaining an average blood pressure of 121/75 millimeters of Mercury (mmHg) — systolic/diastolic pressures, shown with symbols XXXXXX. This is the normal 'intact' subject which represents a relatively healthy young adult male at rest weighing about 70 Kg at rest. His total blood volume is 5 litres (about 9.2 pints), and this is approximately equal to the volume pumped through the heart every minute — the cardiac output. We now simulate the effect of a sudden loss of blood of about 2 litres (about 3.7 pints). MACMAN goes into an immediate state of shock (Figure B), and his blood pressure falls to about 40/21 mmHg. It is noted that the heart is not getting enough blood to the kidneys. Various protective reflexes are immediately invoked, including an increase in heart rate, and at the end of one minute you will notice that the heart rate has risen to 132 beats per minute! His blood pressure recovered slightly to a mean value of 50/34 mmHg. But all is not well. He is "not feeling too well" and "his eyes are going dim." If the simulation were allowed to continue for another three minutes, MACMAN would die and we would have a post-mortem report. The first task, therefore, is to elevate and stabilize his blood pressure. You would try an injection of noradrenalin to constrict his arteries (simulated in MACMAN by doubling the arterial resistance). Figure C shows the effect of this drug; his blood pressure has indeed gone up to 97/70 mmHg and his heart has slowed somewhat to 110 beats per minute. MACMAN is "feeling better but not right yet" — he has stabilized and his symptoms have improved. But note that his blood volume is still at the reduced level of 3 litres and his heart is pumping only about 2.1 litres of blood per minute (cardiac output) — not enough to provide the body cells with the necessary nutrients to maintain a healthy state. However, we shall have mercy on the poor soul and relieve his agony by giving him a blood transfusion of 3 units or about 1.6 litres (and at the same time let the arteries return to their normal state). This is shown in Figure D — after an initial imbalance, the blood pressure and heart rate return to very nearly the normal values (shown in Figure A). The heart is pumping away at 4.6 litres per minute and MACMAN is tickled pink — he acknowledges your efforts and is wondering what nasty motives you have next! This example would have given you an idea of how MACMAN can oblige by letting you perform drastic experiments on him and is even willing to "die" many times for the sake of science! Hundreds of similar experiments can be performed by changing the five "factors" in different combinations. These factors are:

1. Arterial Resistance (expressed as percent of normal value) — is a measure of the pressure required to drive blood through the circulation channels in the body.

2. Venous Resistance (expressed as percent of normal value) — determines the return of blood to the heart.

3. Contractility (expressed as percent of normal value) describes the effectiveness of the heart as a pump.

4. Intra-thoracic pressure (mm of Mercury) is the pressure difference between the pressure inside the chest and the atmospheric pressure.

5. Blood volume is the total blood volume in the body in milli-litres.

MACMAN proved to be a great success and was quickly included in the medical curriculum at McMaster University, and later at many other medical schools around the world. This was easy because, unlike many physiological simulations, the **interface** between the user and the model has been accorded as much attention as the model itself. So anyone who can type with one finger can tinker with MACMAN's physiology. It also led Dr. Dickinson, and his associates in Canada and England, to explore the pos-

```
DO YOU WANT TO  1.CHANGE, 2.CONTINUE, 3.RESTART, 4.STOP THE MODEL
?1
..1.CHANGE FACTORS, 2.BARORECEPTOR FUNCTION, 3.DISPLAYS
?1
TYPE THE NUMBERS OF FACTORS (1-5) TO BE CHANGED
?5
FACTOR  5 (CURRENTLY= 5000.0 ), SPECIFY NEW VALUE
?3000
DO YOU WANT TO  1.CHANGE, 2.CONTINUE, 3.RESTART, 4.STOP THE MODEL
?2

TIME 0  20  40  60  80  100 120 140 160 180 200 220 240 260 280 300
SECS .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
   4      XXXX    *
   8      XXXXX    *
  12       XXXX      *
  16       XXXX        *
  20       XXXXX         *
  24       XXXXX           *
  28        XXXX            *
  32       XXXXX              *
  36        XXXX               *
  40        XXXX               *
  44        XXXX                 *
  48        XXXX                 *
  52        XXXX                 *
  56        XXXX                 *
  60       XXXXX                 *

   I FEEL REALLY TERRIBLE

   I FEEL AS IF I AM GOING TO DIE. MY EYES ARE GOING DARK.
   WHAT IS HAPPENING TO ME ?
YOUR PATIENT IS PASSING HARDLY ANY URINE

B.P. SYSTOLIC= 50  DIASTOLIC= 34  MEAN=  40.0
RIGHT ATRIAL PRESS.= -.3 MM HG  MEAN CAPILL.PRESS.=     3.9 MM HG
CARDIAC OUTPUT=  1.9 L/MIN, STROKE VOL.=  14.7 ML, HEART RATE= 132.
ART.RESIST.=  19.9   VENOUS RESIST.=  2.2 MM HG/L/MIN
CARDIAC CONTRACILITY=  1.1 L/MIN/MM HG FILLING PRESS.
VALUES FOR THIS RUN WERE...

1..ART. RESISTANCE=  100.0 % OF NORMAL
2..VEN. RESISTANCE=  100.0 % OF NORMAL
3..CARDIAC CONTRACTILITY=  100.0 % OF NORMAL
4..MEAN INTRATHORACIC PRES.=  -2.0 MM HG
5..BLOOD VOLUME=  3000.0 ML
```

Figure B MACMAN has suddenly lost 2 litres of blood. His blood pressure has fallen and his heart is beating very rapidly.

```
DO YOU WANT TO  1.CHANGE, 2.CONTINUE, 3.RESTART, 4.STOP THE MODEL
?1
..1.CHANGE FACTORS, 2.BARORECEPTOR FUNCTION, 3.DISPLAYS
?1
TYPE THE NUMBERS OF FACTORS (1-5) TO BE CHANGED
?1
FACTOR  1 (CURRENTLY=  100.0 ), SPECIFY NEW VALUE
?200

DO YOU WANT TO  1.CHANGE, 2.CONTINUE, 3.RESTART, 4.STOP THE MODEL
?2

TIME 0  20  40  60  80  100 120 140 160 180 200 220 240 260 280 300
SECS .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
  64              XXXXXX     *
  68              XXXXXX       *
  72              XXXXXX     *
  76             XXXXXXX    *
  80             XXXXXX     *
  84             XXXXXXX  *
  88             XXXXXX   *
  92             XXXXXX   *
  96             XXXXXX   *
 100             XXXXXXX *
 104             XXXXXXX *
 108             XXXXXXX *
 112              XXXXXX *
 116             XXXXXXX *
 120              XXXXXX *

   I AM VERY TIRED AND EXHAUSTED

   I DON'T FEEL LIKE DOING ANYTHING

   I FEEL BETTER BUT NOT RIGHT YET.
   CAN'T YOU DO SOMETHING FOR ME ?
YOUR PATIENT IS PASSING HARDLY ANY URINE

B.P. SYSTOLIC= 97  DIASTOLIC= 70  MEAN=  79.1
RIGHT ATRIAL PRESS.= -.5 MM HG  MEAN CAPILL.PRESS.=     4.1 MM HG
CARDIAC OUTPUT=  2.1 L/MIN, STROKE VOL.=  19.2 ML, HEART RATE= 110.
ART.RESIST.=  36.7   VENOUS RESIST.=  2.2 MM HG/L/MIN
CARDIAC CONTRACILITY=  1.4 L/MIN/MM HG FILLING PRESS.
VALUES FOR THIS RUN WERE...

1..ART. RESISTANCE=  200.0 % OF NORMAL
2..VEN. RESISTANCE=  100.0 % OF NORMAL
3..CARDIAC CONTRACTILITY=  100.0 % OF NORMAL
4..MEAN INTRATHORACIC PRES.=  -2.0 MM HG
5..BLOOD VOLUME=  3000.0 ML
```

Figure C The effect of a drug to raise blood pressure is simulated. MACMAN has recovered somewhat, his symptoms have improved.

sibilities of simulating other aspects of human physiology. A flurry of activity has followed at McMaster University and St. Bartholomew's Hospital in London. Four new models have evolved. They are far more complex and versatile than MACMAN and have been named MACPUF (simulation of lungs and respiration), MACPEE (simulation of — you guessed it — kidneys and body fluids), MACBABE (simulation of respiration in a pre-mature baby), and finally a very elegant model MACDOPE (simulation of drug distribution in the body), developed by a medical student Ralph Bloch (who is now a Professor at McMaster).

These simulation models have added a new dimension to medical education and research. Not only have they saved many animal experiments (and the associated pain and mess), but the students can now perform a virtually unlimited number of experiments, repeat them several times, try a different treatment each time (or withhold treatment and study the natural course of events). These models have been presented to a wide medical audience and are being used at over 70 different medical schools around the world. In fact, MACMAN and his cousins speak five different languages.

The programs have all been written in standard Fortran and require interactive access to a suitable computer via ordinary (ASCII compatible) terminals. They have been implemented on a range of minicomputers and we plan to adapt them soon to microcomputers. A Basic version of MACMAN has also been developed. The other models have not been translated to Basic, owing to their complexity and high computation requirements.

The development of the MAC family has been a very rewarding experience

to our group. It has provided us a much better understanding of the human systems and in some cases corrected our thinking. As always happens, designing these simulations has pointed out many grey areas in our knowledge. To the student and researcher, this tool is invaluable as it enables them to experiment and test hypotheses — there is no better way to learn than to perform critical experiments. This development opens yet another exciting possibility — the use of simulation in the clinic. We have already built features in the advanced models to allow the user to specify the age, sex, height and weight of the simulated subject. It is then possible by trial and error to "tune-in" a given patient, and then try different treatment plans. We have had some success in this area, though the procedure is fairly involved. We are currently working on making the models self-adjusting. Thus, one could enter the observed results (such as heart rate, blood pressure, levels of oxygen and carbon dioxide in blood, etc.) and let the model work out the other parameters to match the physical state of the actual patient. It would then be possible to predict the outcome of a treatment for a specific person more accurately. This could be a valuable tool for the future doctor and in the Intensive Care Unit.

We feel that computer simulation in medicine offers a lot of exciting possibilities and we are sure to witness new developments in the coming years — we have only begun to scratch the surface. Who knows, someday, you may dial up your 'family computer' about a problem in the middle of the night and out will come the answer "Take two aspirins and call me in the morning!"

```
DO YOU WANT TO  1.CHANGE, 2.CONTINUE, 3.RESTART, 4.STOP THE MODEL
?1
..1.CHANGE FACTORS, 2.BARORECEPTOR FUNCTION, 3.DISPLAYS
?1
TYPE THE NUMBERS OF FACTORS (1-5) TO BE CHANGED
?5  1
FACTOR  5 (CURRENTLY= 3000.0 ), SPECIFY NEW VALUE
?4600
FACTOR  1 (CURRENTLY=  200.0 ), SPECIFY NEW VALUE
?100

DO YOU WANT TO  1.CHANGE, 2.CONTINUE, 3.RESTART, 4.STOP THE MODEL
?2

TIME 0  20  40  60  80  100 120 140 160 180 200 220 240 260 280 300
SECS .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
 124                 XXXX*XXXX
 128                XXX*XXXXX
 132               XX*XXXXXX
 136               X*XXXXXX
 140              XX*XXXXXX
 144              XX*XXXXXX
 148              XX*XXXXXX
 152              XX*XXXXXX
 156              XX*XXXXXXX
 160              XX*XXXXXX
 164              XX*XXXXXX
 168              XX*XXXXXXX
 172              XX*XXXXXX
 176              XX*XXXXXX
 180              XX*XXXXX

   THAT'S BETTER DOCTOR.....
   BUT ARE YOU GOING TO DO ANY OTHER NASTY THINGS TO ME ?

B.P. SYSTOLIC=109  DIASTOLIC= 71  MEAN=  84.2
RIGHT ATRIAL PRESS.= 1.2 MM HG  MEAN CAPILL.PRESS.=    11.3 MM HG
CARDIAC OUTPUT=  4.6 L/MIN, STROKE VOL.=  55.6 ML, HEART RATE=  83.
ART.RESIST.=  16.6   VENOUS RESIST.=  2.2 MM HG/L/MIN
CARDIAC CONTRACILITY=  1.5 L/MIN/MM HG FILLING PRESS.
VALUES FOR THIS RUN WERE...

1..ART. RESISTANCE=  100.0 % OF NORMAL
2..VEN. RESISTANCE=  100.0 % OF NORMAL
3..CARDIAC CONTRACTILITY=  100.0 % OF NORMAL
4..MEAN INTRATHORACIC PRES.=  -2.0 MM HG
5..BLOOD VOLUME=  4600.0 ML

DO YOU WANT TO  1.CHANGE, 2.CONTINUE, 3.RESTART, 4.STOP THE MODEL
?4
```

Figure D MACMAN is now given some blood and the drug is removed. He is feeling much better.

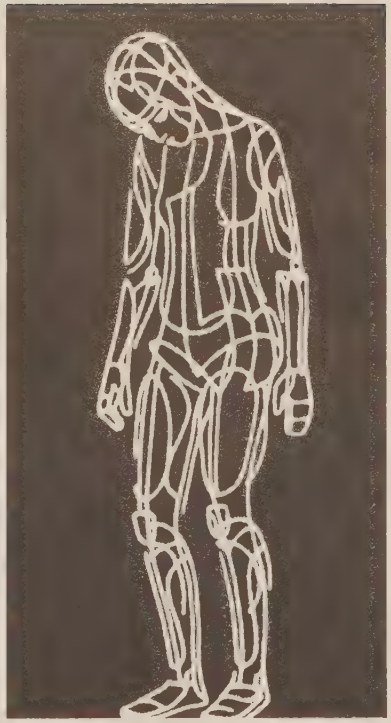
Smokers' Self-Assessment Test

John G. Sparti

This program is based on a smokers self-assessment test, "algorithm." This was written in Basic for the Polymorphic System 8813 with 48K RAM. I have used it in my office and have found that the patients are able to follow directions and are fascinated with the hard copy output. I am making this available to all doctors who have access to a mini, or personal computer. I feel it is a good medical application for in-office use, and I believe it helps to deal with a very serious problem in patient care.

I hope you will find the self-assessment test interesting. □

Dr. John G. Sparti, 318 Main St., Dallas, GA 30132.



Reprinted with permission from *Physician's Microcomputer Report.*

```
10 ON ERROR GOTO 2590
20 REM "WHY SMOKE" ORIGINAL MATERIAL
30 REM BY ARDEN G. CHRISTEN D.D.S.
40 REM AND KENNETH H. COOPER M.D. "STRATEGIC WITHDRAWAL FROM
50 REM CIGARETTE SMOKING" MARCH/APRIL  1979
60 REM FROM "CA-A CANCER JOURNAL FOR CLINICIANS"
70 REM BY JOHN G. SPARTI D.O. & ROGER FORD  BASIC XXX  5/16/79
80 REM POLYMORPHIC SYSTEMS 8813 & 88/MS - DECWRITER II
90 REM 318 MAIN/DALLAS, GA. 30132  :  404-445-5555
100 REM THIS PROGRAM IS TO BE USED FREELY BY ALL (PUBLIC DOMAIN)
110 REM NO GUARANTEE IMPLIED IN THE USE OF THIS MATERIAL!
120 PRINT CHR$(12)
130 PRINT "            WHY DO YOU SMOKE? "
140 PRINT "HERE ARE SOME STATEMENTS MADE BY PEOPLE TO"
150 PRINT "DESCRIBE WHAT THEY GET OUT OF SMOKING CIGARETTES."
160 PRINT "HOW OFTEN DO YOU FEEL THIS WAY ";
170 PRINT "WHEN ARE YOU SMOKING THEM?"
180 PRINT "TO ANSWER THE UPCOMING QUESTIONS, USE THE FOLLOWING"
190 PRINT "NUMBERS. ALWAYS=5  FREQUENTLY=4  OCCASIONALLY=3"
200 PRINT "SELDOM=2 NEVER=1   (THEN HIT RETURN)."
210 PRINT  "(Note:The Above will  appear  on every question)"
220 PRINT "(IMPORTANT PLEASE ANSWER EVERY QUESTION)."
230 PRINT "(TYPE ANY 'LETTER' TO CONTINUE)",
240 WAIT
250 GOSUB 2640
260 PRINT "I SMOKE CIGARETTES IN ORDER TO KEEP MYSELF FROM"
270 PRINT "SLOWING DOWN",\GOSUB 2670\A=Z1
280 GOSUB 2640
290 PRINT "HANDLING A CIGARETTE IS PART OF THE ENJOYMENT"
300 PRINT "OF SMOKING IT",\GOSUB 2670\B=Z1
310 GOSUB 2520
320 PRINT "SMOKING CIGARETTES IS PLEASANT AND RELAXING ",
330 GOSUB 2670\C=Z1
340 GOSUB 2640
350 PRINT "I LIGHT UP A CIGARETTE WHEN I FEEL ANGRY"
360 PRINT "ABOUT SOMETHING",\GOSUB 2670\D=Z1
370 GOSUB 2640
380 PRINT "WHEN I HAVE RUN OUT OF CIGARETTES I FIND IT"
390 PRINT "ALMOST UNBEARABLE UNTIL I CAN GET THEM",
400 GOSUB 2670\E=Z1
410 GOSUB 2640
420 PRINT "I SMOKE CIGARETTES AUTOMATICALLY WITHOUT"
430 PRINT "EVEN BEING AWARE OF IT",\GOSUB 2670\F=Z1
440 GOSUB 2640
450 PRINT "I SMOKE CIGARETTES TO STIMULATE ME, TO PERK"
460 PRINT "MYSELF UP",\GOSUB 2670\G=Z1
470 GOSUB 2640
480 PRINT "PART OF THE ENJOYMENT OF SMOKING A CIGARETTE"
490 PRINT "COMES FROM THE STEPS I TAKE TO LIGHT UP",
500 GOSUB 2670\H=Z1
510 GOSUB 2640
520 PRINT "I FIND CIGARETTES PLEASURABLE",\GOSUB 2670\I=Z1
530 GOSUB 2640
540 PRINT "WHEN I FEEL UNCOMFORTABLE OR UPSET ABOUT"
550 PRINT "SOMETHING I LIGHT UP A CIGARETTE",\GOSUB 2670\J=Z1
560 GOSUB 2640
570 PRINT "I AM VERY MUCH AWARE OF THE FACT WHEN I AM NOT"
580 PRINT "SMOKING A CIGARETTE",\GOSUB 2670\K=Z1
590 GOSUB 2640
600 PRINT "I LIGHT UP A CIGARETTE WITHOUT REALIZING I STILL"
610 PRINT "HAVE ONE BURNING IN THE ASHTRAY",\GOSUB 2670\L=Z1
620 GOSUB 2640
630 PRINT "I SMOKE CIGARETTES TO GIVE ME A 'LIFT'",\GOSUB 2670
640 M=Z1
650 GOSUB 2640
660 PRINT "WHEN I SMOKE A CIGARETTE, PART OF THE ENJOYMENT"
670 PRINT "IS WATCHING THE SMOKE AS I EXHALE IT",
680 GOSUB 2670\N=Z1
690 GOSUB 2640
700 PRINT "I WANT A CIGARETTE MOST WHEN I AM COMFORTABLE"
710 PRINT "AND RELAXED",\GOSUB 2670\O=Z1
720 GOSUB 2640
730 PRINT "WHEN I FEEL 'BLUE' OR WANT TO TAKE MY MIND OFF"
740 PRINT "CARES AND WORRIES, I SMOKE CIGARETTES",
750 GOSUB 2670\P=Z1
760 GOSUB 2640
770 PRINT "I GET A REAL GNAWING HUNGER FOR A CIGARETTE"
780 PRINT "WHEN I HAVEN'T SMOKED FOR A WHILE",
790 GOSUB 2670\Q=Z1
800 GOSUB 2640
810 PRINT "I'VE FOUND A CIGARETTE IN MY MOUTH AND DIDN'T"
820 PRINT "REMEMBER PUTTING IT THERE",\GOSUB 2670\R=Z1
830 S=A+G+M\H1=B+H+N
840 P1=C+I+O\C1=D+J+P
850 C2=E+K+Q\H2=F+L+R
860 PRINT CHR$ (12)
870 PRINT "SCORES OF 11 OR ABOVE INDICATE THAT THIS FACTOR IS"
880 PRINT "AN IMPORTANT SOURCE OF SATISFACTION FOR THE SMOKER."
890 PRINT "SCORES OF 7 OR LESS ARE LOW AND PROBABLY INDICATE"
900 PRINT "THAT THIS FACTOR DOES NOT APPLY TO YOU."
910 PRINT "SCORES IN BETWEEN ARE MARGINAL."
920 PRINT
930 IF S>=10.5  THEN PRINT "YOUR STIMULATION SCORE IS ",S
940 IF H1>=10.5 THEN PRINT "YOUR HANDLING SCORE IS ",H1
```

```
950 IF P1<10.5 THEN 970
960 PRINT 'YOUR PLEASURABLE RELAXATION SCORE IS ',P1
970 IF C1<10.5 THEN 990
980 PRINT 'YOUR CRUTCH: TENSION  REDUCTION SCORE IS ',C1
990 IF C2<10.5 THEN 1010
1000 PRINT 'YOUR CRAVING: PSYCHOLOGICAL ADDICTION SCORE IS ',C2
1010 IF H2>=10.5 THEN PRINT 'YOUR HABIT SCORE IS ',H2
1020 FILE:2,LIST
1030 PRINT:2,CHR$(12),TAB(30),"*** Report ***"
1040 PRINT:2,'YOUR STIMULATION SCORE IS ',S
1050 PRINT:2,'YOUR HANDLING SCORE IS ',H1
1060 PRINT:2,'YOUR PLEASURE RELAXATION SCORE IS ',P1
1070 PRINT:2,'YOUR CRUTCH: TENSION REDUCTION SCORE IS ',C1
1080 PRINT:2,'YOUR CRAVING: PSYCHOLOGICAL ',
1090 PRINT:2,'ADDICTION SCORE IS ',C2
1100 PRINT:2,'YOUR HABIT SCORE IS ',H2
1110 PRINT:2,' '
1120 PRINT:2,'SCORES OF 11 OR ABOVE INDICATE THAT THIS FACTOR'
1130 PRINT:2,'IS AN IMPORTANT SOURCE OF SATISFACTION FOR YOU'
1140 PRINT:2,'THE SMOKER.SCORES BETWEEN 7 AND 11 ARE MARGINAL'
1150 PRINT:2,'SCORES OF 7 OR LESS INDICATE THAT THESE FACTORS'
1160 PRINT:2,'PROBABLY DO NOT APPLY TO YOU.'
1170 PRINT:2,' '
1180 IF S<7.5 THEN 1240
1190 IF S<10.5 THEN 1220
1200 PRINT:2,'STIMULATION IS AN IMPORTANT SOURCE OF'
1210 GOTO 1230
1220 PRINT:2,'STIMULATION IS A MARGINAL SOURCE OF'
1230 PRINT:2,'SATISFACTION FOR YOU.'\GOTO 1270
1240 PRINT:2,'SINCE STIMULATION IS NOT AN IMPORTANT FACTOR TO'
1250 PRINT:2,'YOU, THE FOLLOWING PARAGRAPH PROBABLY DOESN'T'
1260 PRINT:2,'APPLY TO YOU.'
1270 PRINT:2,'STIMULATION (10% OF SMOKERS) THIS TYPE OF SMOKER'
1280 PRINT:2,'IS STIMULATED BY THE CIGARETTE.  IT HELPS HIM TO'
1290 PRINT:2,'WAKE UP IN THE MORNING, TO ORGANIZE HIS ENERGIES'
1300 PRINT:2,'AND TO KEEP HIM GOING.  MANY SMOKERS  REPORT'
1310 PRINT:2,'THAT WHILE SMOKING THEY EXPERIENCE A SHARPENING'
1320 PRINT:2,'OF INTELECTUAL CAPACITY AND AN INCREASE IN'
1330 PRINT:2,'IMPULSE CONTROL.'
1340 PRINT:2,' '
1350 IF H1<7.5 THEN 1410
1360 PRINT:2,'IF H1<10.5 THEN 2710
1370 PRINT:2,'HANDLING IS AN IMPORTANT SOURCE OF SATISFACTION'
1380 PRINT:2,'FOR YOU.'\GOTO 1440
1390 PRINT:2,'HANDLING IS A MARGINAL SOURCE OF SATISFACTION'
1400 PRINT:2,'FOR YOU.'\GOTO 1440
1410 PRINT:2,'SINCE HANDLING IS NOT AN IMPORTANT FACTOR TO'
1420 PRINT:2,'YOU, THE FOLLOWING PARAGRAPH PROBABLY DOESN'T'
1430 PRINT:2,'APPLY TO YOU.'
1440 PRINT:2,'HANDLING: SENSORIMOTOR MANIPULATION (10%)'
1450 PRINT:2,'THIS TYPE ENJOYS MANIPULATING THE CIGARETTE'
1460 PRINT:2,'WITH HIS HANDS, WATCHING THE SMOKE CURL AS HE'
1470 PRINT:2,'EXHALES, AND MAKING A PRODUCTION OF LIGHTING'
1480 PRINT:2,'THE CIGARETTE, HOLDING IT, AND FLICKING THE'
1490 PRINT:2,'ASHES.'
1500 PRINT:2,' '
1510 IF P1<7.5 THEN 1570
1520 IF P1<10.5 THEN 1550
1530 PRINT:2,'PLEASUREABLE RELAXATION IS AN IMPORTANT SOURCE'
1540 PRINT:2,'OF SATISFACTION FOR YOU.'\GOTO 1600
1550 PRINT:2,'PLEASUREABLE RELAXATION IS A MARGINAL SOURCE OF'
1560 PRINT:2,'SATISFACTION FOR YOU.'\GOTO 1600
1570 PRINT:2,'SINCE PLEASURABLE RELAXATION IS NOT AN IMPORTANT'
1580 PRINT:2,'FACTOR TO YOU, THE FOLLOWING PARAGRAPH PROBABLY'
1590 PRINT:2,'DOESN'T APPLY TO YOU.'
1600 PRINT:2,'PLEASUREABLE RELAXATION (15%).'
1610 PRINT:2,'THIS SMOKER GETS REAL, HONEST  PLEASURE FROM'
1620 PRINT:2,'SMOKING, ESPECIALLY AFTER A DINNER OR WITH A'
1630 PRINT:2,'COCKTAIL.  HE TENDS TO SMOKE TO ACCENTUATE OR'
1640 PRINT:2,'ENHANCE PLEASURABLE FEELINGS ACCOMPANYING A'
1650 PRINT:2,'STATE OF WELL BEING.'
1660 PRINT:2,' '
1670 PRINT:2,'THE ABOVE SMOKERS CHARACTERISTICALLY SMOKE '
1680 PRINT:2,'UNDER PLEASANT CIRCUMSTANCES THAT ARE RELAXING.'
1690 PRINT:2,'THEY USUALLY REQUIRE AN ADEQUATE SUBSTITUTE FOR'
1700 PRINT:2,'QUITTING AND OFTEN FIND THAT IT IS NOT TOO '
1710 PRINT:2,'DIFFICULT TO STOP.'
1720 PRINT:2,' '
1730 IF C1<7.5 THEN 1790
1740 IF C1<10.5 THEN 1770
1750 PRINT:2,'CRUTCH:TENSION REDUCTION IS AN IMPORTANT SOURCE'
1760 PRINT:2,'OF SATISFACTION FOR YOU.'\GOTO 1820
1770 PRINT:2,'CRUTCH: TENSION REDUCTION IS A MARGINAL SOURCE'
1780 PRINT:2,'OF SATISFACTION FOR YOU.'\GOTO 1820
1790 PRINT:2,'SINCE CRUTCH: TENSION REDUCTION IS NOT AN'
1800 PRINT:2,'IMPORTANT FACTOR TO YOU, THE FOLLOWING PARAGRAPH'
1810 PRINT:2,'PROBABLY DOESN'T APPLY TO YOU.'
1820 PRINT:2,'CRUTCH: TENSION REDUCTION (30%).'
1830 PRINT:2,'THIS NEGATIVE-AFFECT TYPE OF SMOKER USES'
1840 PRINT:2,'CIGARETTES FOR THE SEDATIVE OR TRANQUILIZER'
1850 PRINT:2,'EFFECT IN MOMENTS OF STRESS, FEAR, SHAME,'
1860 PRINT:2,'DISGUST, DISCOMFORT, OR PRESSURE.  HE USES'
1870 PRINT:2,'CIGARETTES TO HELP HIM COPE WITH PROBLEMS.'
1880 PRINT:2,'SUBSTITUTIONS WILL GENERALLY NOT HELP THIS'
1890 PRINT:2,'SMOKER, AND SUCCESS IS GAINED FROM LEARNING'
1900 PRINT:2,'HOW TO PROPERLY MANAGE THE SITUATIONS THAT'
1910 PRINT:2,'PRODUCE THE BAD FEELINGS IN THE FIRST PLACE.'
1920 PRINT:2,' '
1930 IF C2<7.5 THEN 1990
1940 IF C2<10.5 THEN 1970
1950 PRINT:2,'CRAVING: PSYCHOLOGICAL ADDICTION IS AN IMPORTANT'
1960 GOTO 1980
1970 PRINT:2,'CRAVING: PSYCHOLOGICAL ADDICTION IS A MARGINAL'
1980 PRINT:2,'SOURCE OF SATISFACTION FOR YOU.'\GOTO 2020
1990 PRINT:2,'SINCE CRAVING: PSYCHOLOGICAL ADDICTION IS NOT AN'
2000 PRINT:2,'IMPORTANT FACTOR TO YOU, THE FOLLOWING PARAGRAPH'
2010 PRINT:2,'PROBABLY DOESN'T APPLY TO YOU.'
2020 PRINT:2,'CRAVING: PSYCHOLOGICAL ADDICTION (25%). '
2030 PRINT:2,'THIS TYPE OF SMOKER FEELS TOTALLY DEPENDENT'
2040 PRINT:2,'UPON HIS CIGARETTES, AND ALTERNATES BETWEEN AN'
2050 PRINT:2,'INCREASE OF POSITIVE AND A DECREASE OF NEGATIVE'
2060 PRINT:2,'FEELINGS.  HE IS CONSTANTLY AWARE WHEN HE IS NOT'
2070 PRINT:2,'SMOKING AND BEGINS THE CRAVING FOR THE NEXT'
2080 PRINT:2,'CIGARETTE WHEN HE PUTS OUT THE PRESENT ONE.'
2090 PRINT:2,'TAPERING OFF DOESN'T SEEM TO WORK; THE ONLY'
2100 PRINT:2,'SOLUTION IS TO QUIT `COLD TURKEY'.'
2110 PRINT:2,' '
2120 IF H2<7.5 THEN 2180
2130 IF H2<10.5 THEN 2160
2140 PRINT:2,'HABIT IS AN IMPORTANT SOURCE OF'
2150 GOTO 2170
2160 PRINT:2,'HABIT IS A MARGINAL SOURCE OF'
2170 PRINT:2,'SATISFACTION FOR YOU.'\GOTO 2210
2180 PRINT:2,'SINCE HABIT IS NOT AN IMPORTANT FACTOR TO'
2190 PRINT:2,'YOU, THE FOLLOWING PARAGRAPH PROBABLY DOESN'T'
2200 PRINT:2,'APPLY TO YOU.'
2210 PRINT:2,'HABIT (10%).'
2220 PRINT:2,'THE HABITUAL SMOKER GETS VERY LITTLE',
2230 PRINT:2,' SATISFACTION'
2240 PRINT:2,'FROM HIS HABIT AND PERFORMS IT SOMEWHAT'
2250 PRINT:2,'AUTOMATICALLY.  HE MAY SMOKE A CIGARETTE WITHOUT'
2260 PRINT:2,'REALIZING IT OR EVEN WANTING IT.  THERE IS VERY'
2270 PRINT:2,'LITTLE EMOTIONAL COMPONENT TO HIS SMOKING'
2280 PRINT:2,'ACTIVITY AND A  MINIMAL AMOUNT OF AWARENESS.'
2290 PRINT:2,'IT IS IMPORTANT FOR HIM TO DEVELOP THIS'
2300 PRINT:2,'AWARENESS AND TO UNDERSTAND THE PATTERN OF HIS'
```

```
2310 PRINT:2,"SMOKING."
2320 PRINT:2,"  "
2330 PRINT:2,"CIGARETTE SMOKING CAN BE VIEWED MAINLY AS A"
2340 PRINT:2,"PSYCHOLOGICAL HABIT WITH ACCOMPANYING"
2350 PRINT:2,"PHARMACOLOGICAL AND SOCIAL SATISFACTIONS AND"
2360 PRINT:2,"REINFORCEMENTS.  IT IS MAINLY A LEARNED BEHAVIOR"
2370 PRINT:2,"OCCURRING IN RESPONSE TO DIRECT AND INDIRECT"
2380 PRINT:2,"SOCIAL PRESSURES.  THOUGH EACH PERSON'S SMOKING"
2390 PRINT:2,"HABIT IS INDIVIDUALIZED, STUDIES SHOW THAT"
2400 PRINT:2,"SMOKERS FIND AT LEAST SIX PSYCHOLOGICAL"
2410 PRINT:2,"SATISFACTIONS IN THE HABIT: 1. STIMULATION"
2420 PRINT:2,"2. HANDLING  3. RELAXATION 4. TENSION REDUCTION"
2430 PRINT:2,"5. CRAVING 6. HABIT"
2440 PRINT:2,"  "
2450 PRINT:2,"TO QUIT SMOKING THE INDIVIDUAL MUST UNDERGO"
2460 PRINT:2,"PHYSICAL AND PSYCHOLOGICAL WITHDRAWAL.  PHYSICAL"
2470 PRINT:2,"WITHDRAWAL SYMPTOMS LAST ABOUT ONE WEEK BUT IT"
2480 PRINT:2,"MAY TAKE MONTHS OR YEARS TO RESTRUCTURE A LIFE"
2490 PRINT:2,"WITHOUT CIGARETTES.  QUITTING IS A LONG-TERM"
2500 PRINT:2,"LEARNING PROCESS INVOLVING A GREAT DEAL OF"
2510 PRINT:2,"UNCONSCIOUS, CONSTRUCTIVE AND PAINFUL CONFLICT."
2520 PRINT:2,"ONE CAN STOP SMOKING IF THE PATIENT WISHES TO"
2530 PRINT:2,"DO SO, ATTACKS THE PROBLEM ON A BROAD FRONT,"
2540 PRINT:2,"DOESN'T LOOK BACK, AND IF SUITABLE SATISFYING"
2550 PRINT:2,"ALTERNATIVES AND INVOLVEMENTS ARE DISCOVERED."
2560 PRINT:2,CHR$(12)
2570 REM  END  OF PROGRAM
2580 STOP
2590 PRINT "YOU ENTERED DATA IN ERROR!"
2600 RESET(ERR)\GOTO 2670
2610 REM END OF ERROR ROUTINE
2620 STOP
2630 REM Answer Routine
2640 PRINT CHR$(12)
2650 PRINT "ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1"
2660 RETURN
2670 INPUT " ?",Z1
2680 IF Z1>=1 AND Z1<=5 AND Z1=INT(Z1) THEN RETURN
2690 PRINT "Answer NOT valid!!!"\GOTO 2670
```

```
A
   270    830

B
   300    830

C1
   840    970    980   1070   1730   1740

C2
   850    990   1000   1090   1930   1940

C
   330    840

D
   360    840

E
   400    850

F
   430    850

G
   460    830

H1
   830    940   1050   1350

H2
   850   1010   1100   2120   2130

H
   500    830

I
   520    840

J
   550    840

K
   580    850

L
   610    850

M
   640    830

N
   680    830

O
   710    840

P1
   840    950    960   1060   1510   1520

P
   750    840

Q
   790    850

R
   820    850   1030

S
   830    930   1040   1180   1190

Z1
   270    300    330    360    400    430    460    500    520
   550    580    610    640    680    710    750    790    820
  2670   2680
```

```
                WHY DO YOU SMOKE?
HERE ARE SOME STATEMENTS MADE BY PEOPLE TO
DESCRIBE WHAT THEY GET OUT OF SMOKING CIGARETTES.
HOW OFTEN DO YOU FEEL THIS WAY WHEN YOU SMOKING THEM?
TO ANSWER THE UPCOMING QUESTIONS, USE THE FOLLOWING
NUMBERS. ALWAYS=5  FREQUENTLY=4  OCCASIONALLY=3
SELDOM=2 NEVER=1   (THEN HIT RETURN).
(Note:The Above will  appear  on every question)
(IMPORTANT PLEASE ANSWER EVERY QUESTION).
(TYPE ANY 'LETTER' TO CONTINUE)Waiting...
```

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I SMOKE CIGARETTES IN ORDER TO KEEP MYSELF FROM SLOWING DOWN ?2

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
HANDLING A CIGARETTE IS PART OF THE ENJOYMENT OF SMOKING IT ?1†
YOU ENTERED DATA IN ERROR!
?1
SMOKING CIGARETTES IS PLEASANT AND RELAXING ?4

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I LIGHT UP A CIGARETTE WHEN I FEEL ANGRY ABOUT SOMETHING ?2

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
WHEN I HAVE RUN OUT OF CIGARETTES I FIND IT ALMOST UNBEARABLE UNTIL I CAN GET THEM ?5

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I SMOKE CIGARETTES AUTOMATICALLY WITHOUT EVEN BEING AWARE OF IT ?3

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I SMOKE CIGARETTES TO STIMULATE ME, TO PERK MYSELF UP ?5

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
PART OF THE ENJOYMENT OF SMOKING A CIGARETTE COMES FROM THE STEPS I TAKE TO LIGHT UP ?4

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I FIND CIGARETTES PLEASURABLE ?5

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
WHEN I FEEL UNCOMFORTABLE OR UPSET ABOUT SOMETHING I LIGHT UP A CIGARETTE ?4

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I AM VERY MUCH AWARE OF THE FACT WHEN I AM NOT SMOKING A CIGARETTE ?3

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I LIGHT UP A CIGARETTE WITHOUT REALIZING I STILL HAVE ONE BURNING IN THE ASHTRAY ?4

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I SMOKE CIGARETTES TO GIVE ME A 'LIFT' ?4

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
WHEN I SMOKE A CIGARETTE, PART OF THE ENJOYMENT IS WATCHING THE SMOKE AS I EXHALE IT ?1

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I WANT A CIGARETTE MOST WHEN I AM COMFORTABLE AND RELAXED ?5

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
WHEN I FEEL 'BLUE' OR WANT TO TAKE MY MIND OFF CARES AND WORRIES, I SMOKE CIGARETTES ?4

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I GET A REAL GNAWING HUNGER FOR A CIGARETTE WHEN I HAVEN'T SMOKED FOR A WHILE ?5

ALWAYS=5, FREQ=4, OCCAS=3, SELDOM=2, NEV=1
I'VE FOUND A CIGARETTE IN MY MOUTH AND DIDN'T REMEMBER PUTTING IT THERE ?3

SCORES OF 11 OR ABOVE INDICATE THAT THIS FACTOR IS AN IMPORTANT SOURCE OF SATISFACTION FOR THE SMOKER.
SCORES OF 7 OR LESS ARE LOW AND PROBABLY INDICATE THAT THIS FACTOR DOES NOT APPLY TO YOU.
SCORES IN BETWEEN ARE MARGINAL.

YOUR STIMULATION SCORE IS 11
YOUR PLEASURABLE RELAXATION SCORE IS 14
YOUR CRAVING: PSYCHOLOGICAL ADDICTION SCORE IS 13
YOUR STIMULATION SCORE IS 11
YOUR HANDLING SCORE IS 6
YOUR PLEASURE RELAXATION SCORE IS 14
YOUR CRUTCH: TENSION REDUCTION SCORE IS 10
YOUR CRAVING: PSYCHOLOGICAL ADDICTION SCORE IS 13
YOUR HABIT SCORE IS 10

SCORES OF 11 OR ABOVE INDICATE THAT THIS FACTOR IS AN IMPORTANT SOURCE OF SATISFACTION FOR YOU THE SMOKER.SCORES BETWEEN 7 AND 11 ARE MARGINAL SCORES OF 7 OR LESS INDICATE THAT THESE FACTORS PROBABLY DO NOT APPLY TO YOU.

STIMULATION IS AN IMPORTANT SOURCE OF SATISFACTION FOR YOU.
STIMULATION (10% OF SMOKERS) THIS TYPE OF SMOKER IS STIMULATED BY THE CIGARETTE. IT HELPS HIM TO WAKE UP IN THE MORNING, TO ORGANIZE HIS ENERGIES AND TO KEEP HIM GOING. MANY SMOKERS REPORT THAT WHILE SMOKING THEY EXPERIENCE A SHARPENING OF INTELECTUAL CAPACITY AND AN INCREASE IN IMPULSE CONTROL.

SINCE HANDLING IS NOT AN IMPORTANT FACTOR TO YOU, THE FOLLOWING PARAGRAPH PROBABLY DOESN'T APPLY TO YOU.
HANDLING: SENSORIMOTOR MANIPULATION (10%)
THIS TYPE ENJOYS MANIPULATING THE CIGARETTE WITH HIS HANDS, WATCHING THE SMOKE CURL AS HE EXHALES, AND MAKING A PRODUCTION OF LIGHTING THE CIGARETTE, HOLDING IT, AND FLICKING THE ASHES.

PLEASUREABLE RELAXATION IS AN IMPORTANT SOURCE OF SATISFACTION FOR YOU.
PLEASUREABLE RELAXATION (15%).
THIS SMOKER GETS REAL, HONEST PLEASURE FROM SMOKING, ESPECIALLY AFTER A DINNER OR WITH A COCKTAIL. HE TENDS TO SMOKE TO ACCENTUATE OR ENHANCE PLEASURABLE FEELINGS ACCOMPANYING A STATE OF WELL BEING.

THE ABOVE SMOKERS CHARACTERISTICALLY SMOKE UNDER PLEASANT CIRCUMSTANCES THAT ARE RELAXING. THEY USUALLY REQUIRE AN ADEQUATE SUBSTITUTE FOR QUITTING AND OFTEN FIND THAT IT IS NOT TOO DIFFICULT TO STOP.

CRUTCH: TENSION REDUCTION IS A MARGINAL SOURCE OF SATISFACTION FOR YOU.
CRUTCH: TENSION REDUCTION (30%).
THIS NEGATIVE-AFFECT TYPE OF SMOKER USES CIGARETTES FOR THE SEDATIVE OR TRANQUILIZER EFFECT IN MOMENTS OF STRESS, FEAR, SHAME, DISGUST, DISCOMFORT, OR PRESSURE. HE USES CIGARETTES TO HELP HIM COPE WITH PROBLEMS. SUBSTITUTIONS WILL GENERALLY NOT HELP THIS SMOKER, AND SUCCESS IS GAINED FROM LEARNING HOW TO PROPERLY MANAGE THE SITUATIONS THAT PRODUCE THE BAD FEELINGS IN THE FIRST PLACE.

CRAVING: PSYCHOLOGICAL ADDICTION IS AN IMPORTANT SOURCE OF SATISFACTION FOR YOU.
CRAVING: PSYCHOLOGICAL ADDICTION (25%).
THIS TYPE OF SMOKER FEELS TOTALLY DEPENDENT UPON HIS CIGARETTES, AND ALTERNATES BETWEEN AN INCREASE OF POSITIVE AND A DECREASE OF NEGATIVE FEELINGS. HE IS CONSTANTLY AWARE WHEN HE IS NOT SMOKING AND BEGINS THE CRAVING FOR THE NEXT CIGARETTE WHEN HE PUTS OUT THE PRESENT ONE. TAPERING OFF DOESN'T SEEM TO WORK; THE ONLY SOLUTION IS TO QUIT 'COLD TURKEY'.

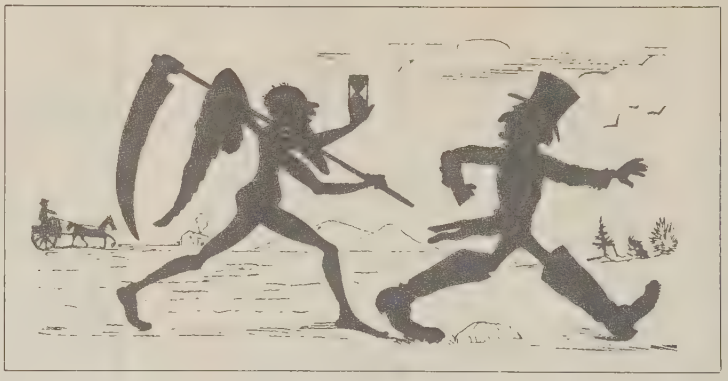
HABIT IS A MARGINAL SOURCE OF SATISFACTION FOR YOU.
HABIT (10%).
THE HABITUAL SMOKER GETS VERY LITTLE SATISFACTION FROM HIS HABIT AND PERFORMS IT SOMEWHAT AUTOMATICALLY. HE MAY SMOKE A CIGARETTE WITHOUT REALIZING IT OR EVEN WANTING IT. THERE IS VERY LITTLE EMOTIONAL COMPONENT TO HIS SMOKING ACTIVITY AND A MINIMAL AMOUNT OF AWARENESS. IT IS IMPORTANT FOR HIM TO DEVELOP THIS AWARENESS AND TO UNDERSTAND THE PATTERN OF HIS SMOKING.

CIGARETTE SMOKING CAN BE VIEWED MAINLY AS A PSYCHOLOGICAL HABIT WITH ACCOMPANYING PHARMACOLOGICAL AND SOCIAL SATISFACTIONS AND REINFORCEMENTS. IT IS MAINLY A LEARNED BEHAVIOR OCCURRING IN RESPONSE TO DIRECT AND INDIRECT SOCIAL PRESSURES. THOUGH EACH PERSON'S SMOKING HABIT IS INDIVIDUALIZED, STUDIES SHOW THAT SMOKERS FIND AT LEAST SIX PSYCHOLOGICAL SATISFACTIONS IN THE HABIT: 1. STIMULATION 2. HANDLING 3. RELAXATION 4. TENSION REDUCTION 5. CRAVING 6. HABIT

TO QUIT SMOKING THE INDIVIDUAL MUST UNDERGO PHYSICAL AND PSYCHOLOGICAL WITHDRAWAL. PHYSICAL WITHDRAWAL SYMPTOMS LAST ABOUT ONE WEEK BUT IT MAY TAKE MONTHS OR YEARS TO RESTRUCTURE A LIFE WITHOUT CIGARETTES. QUITTING IS A LONG-TERM LEARNING PROCESS INVOLVING A GREAT DEAL OF UNCONSCIOUS, CONSTRUCTIVE AND PAINFUL CONFLICT. ONE CAN STOP SMOKING IF THE PATIENT WISHES TO DO SO, ATTACKS THE PROBLEM ON A BROAD FRONT, DOESN'T LOOK BACK, AND IF SUITABLE SATISFYING ALTERNATIVES AND INVOLVEMENTS ARE DISCOVERED.

Stop in line 2610
>>

Life Expectancy



This program is a life-expectancy test derived from Peter Passell's book "How To." The test asks you a series of questions dealing with your life-style and environment. At the end of the questioning, the program gives your estimated life-expectancy and the percentage of the population you should outlive.

You may wish to experiment with certain variables to see what effect they will have on your lifespan. It's unlikely that you want to change your sex, but you may wish to check out the effect of smoking, drinking, mental attitude or weight.

This program was written by John E. Rogers.

```
RUN
                    LIFE EXPECTANCY
                  CREATIVE COMPUTING
                MORRISTOWN, NEW JERSEY



THIS IS A LIFE EXPECTANCY TEST.
   DO YOU WISH INSTRUCTIONS? YES


   THIS IS A TEST TO PREDICT YOUR LIFE EXPECTANCY.  I
WILL ASK YOU A SERIES OF SHORT QUESTIONS, WHICH YOU WILL
REPLY BY TYPING IN THE CORRESPONDING ANSWER TO THE
QUESTION.

       EXAMPLE:  WHAT IS YOUR SEX?
                 M=MALE
                 F=FEMALE
'M' AND 'F' ARE THE POSSIBLE REPLIES TO THE QUESTION, ANSWER
LIKE THIS:
            CHOOSE ONE OF THE LETTERS ABOVE? M
TYPING AN 'M' SIGNIFIES YOU ARE A MALE.


  +++SEX+++
  ARE YOU MALE OR FEMALE?
  M= MALE.
  F= FEMALE.
CHOOSE ONE OF THE LETTERS ABOVE? M

  +++LIFE STYLE+++
  WHERE DO YOU LIVE?
  G= IF YOU LIVE IN AN URBAN AREA WITH A POPULATION OVER 2 MIL.
  K= IF YOU LIVE IN A TOWN UNDER 10,000, OR ON A FARM.
  I= NEITHER.
CHOOSE ONE OF THE LETTERS ABOVE? I

  HOW DO YOU WORK?
  M= IF YOU WORK BEHIND A DESK.
  L= IF YOUR WORK REQUIRES HEAVY PHYSICAL LABOR.
  I= NONE OF THE ABOVE.
CHOOSE ONE OF THE LETTERS ABOVE? I

  HOW LONG DO YOU EXERCISE STRENUOUSLY,
  (TENNIS, RUNNING, SWIMMING, ETC.)?
  F= FIVE TIMES A WEEK FOR AR LEAST A HALF HOUR.
  K= JUST TWO OR THREE TIMES A WEEK.
  I= DO NOT EXERCISE IN THIS FASHION.
CHOOSE ONE OF THE LETTERS ABOVE? K

  WHO DO YOU LIVE WITH?
  N= IF YOU LIVE WITH A SPOUSE, FRIEND, OR IN A FAMILY.
  H= IF YOU'VE LIVED ALONE FOR 1-10 YEARS SINCE AGE 25.
  G= FOR 11-20 YEARS.
  M= FOR 21-30 YEARS.
  E= FOR 31-40 YEARS.
  M= MORE THAN 40 YEARS.
CHOOSE ONE OF THE LETTERS ABOVE? N

  +++PSYCHE+++
  DO YOU SLEEP MORE THAN 10 HOURS A NIGHT?
  I= NO.
  E=YES.
CHOOSE ONE OF THE LETTERS ABOVE? I

  +++MENTAL STATE+++
  M= IF YOU ARE INTENSE, AGGRESSIVE, OR EASILY ANGERED.
  L= IF YOU ARE EASY GOING, RELAXED, OR A FOLLOWER.
  I= NEITHER.
CHOOSE ONE OF THE LETTERS ABOVE? L

  +++HOW YOU FEEL+++
  ARE YOU HAPPY OR UNHAPPY?
  J= HAPPY.
  G= UNHAPPY.
  I= NEITHER.
CHOOSE ONE OF THE LETTERS ABOVE? J

  +++FACTORS+++
  HAVE YOU HAD A SPEEDING TICKET IN THE LAST YEAR?
  H= YES.
  I=NO.
CHOOSE ONE OF THE LETTERS ABOVE? I

  +++INCOME+++
  DO YOU EARN MORE THAN $50,000 A YEAR?
  G= YES.
  I=NO.
CHOOSE ONE OF THE LETTERS ABOVE? IO

  +++SCHOOLING+++
  J= IF YOU HAVE FINISHED COLLEGE.
  L= IF YOU HAVE FINISHED COLLEGE WITH A GRADUATE
  OR PROFESSIONAL DEGREE.
  I= NOTHING LISTED.
CHOOSE ONE OF THE LETTERS ABOVE? I
```

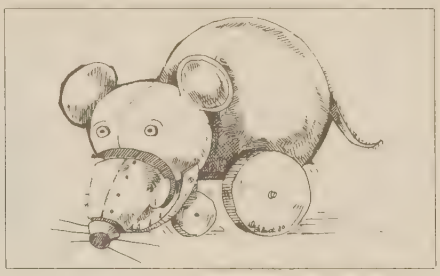
```
   +++AGE+++
   ARE YOU 65 OR OLDER AND STILL WORKING?
   L= YES.
   I= NO.
CHOOSE ONE OF THE LETTERS ABOVE? I

   +++HEREDITY+++
   K= IF ANY GRANDPARENTS LIVED TO 85 YEARS OLD.
   O= IF ALL FOUR GRANDPARENTS LIVED TO 80 YEARS OLD.
   I= NO GRANDPARENTS QUALIFY IN THE ABOVE.
CHOOSE ONE OF THE LETTERS ABOVE? I

   HAS ANY PARENT DIED OF A STROKE OR HEART ATTACK
   BEFORE THE AGE OF 50?
   E= YES.
   I= NO.
CHOOSE ONE OF THE LETTERS ABOVE? E

   +++FAMILY DISEASES+++
   ANY PARENT, BROTHER, OR SISTER UNDER 50 HAS (OR HAD)
   CANCER, A HEART CONDITION, OR DIABETES SINCE CHILDHOOD?
   M= YES.
   I= NO.
CHOOSE ONE OF THE LETTERS ABOVE? I

   +++HEALTH+++
   HOW MUCH DO YOU SMOKE?
   A= IF YOU SMOKE MORE THAN TWO PACKS A DAY.
   C= ONE TO TWO PACKS A DAY.
   M= ONE HALF TO ONE PACK A DAY.
   I= DON'T SMOKE.
CHOOSE ONE OF THE LETTERS ABOVE? I

   +++DRINK+++
   DO YOU DRINK THE EQUIVALENT OF A
   QUARTER BOTTLE OF ALCOHOLIC BEVERAGE A DAY?
   H= YES.
   I= NO.
CHOOSE ONE OF THE LETTERS ABOVE? I

   +++WEIGHT+++
   A= IF YOU ARE OVERWEIGHT BY 50 POUNDS OR MORE.
   E= OVER BY 30-50 POUNDS.
   G= OVER BY 10-30 POUNDS.
   I= NOT OVER WEIGHT.
CHOOSE ONE OF THE LETTERS ABOVE? I

   +++CHECKUPS+++
   DO YOU?  IF YOU ARE A MALE OVER 40 HAVE AN ANNUAL CHECKUP?
   K= YES.
   I= IF NO OR NOT A MALE OR UNDER 40 YEARS OLD.
CHOOSE ONE OF THE LETTERS ABOVE? I

   DO YOU? IF YOU ARE A WOMAN SEE A GYNECOLOGIST ONCE A YEAR?
   K= YES.
   I= IF NO OR NOT A WOMAN.
CHOOSE ONE OF THE LETTERS ABOVE? I

   +++CURRENT AGE+++
   K= IF YOU ARE BETWEEN 30 AND 40 YEARS OLD.
   L= BETWEEN 40 AND 50.
   F= BETWEEN 50 AND 70.
   N= OVER 70.
   I= UNDER 30.
CHOOSE ONE OF THE LETTERS ABOVE? I

YOU ARE EXPECTED TO LIVE TO THE AGE OF 76 YEARS
OUT LIVING 61% OF THE MEN AND 39% OF THE WOMEN.
Ok
```

```
LIST
10 PRINT TAB(21);"LIFE EXPECTANCY"
20 PRINT TAB(20);"CREATIVE COMPUTING"
30 PRINT TAB(18);"MORRISTOWN, NEW JERSEY"
40 PRINT:PRINT:PRINT
270 PRINT :PRINT:PRINT
280 PRINT "THIS IS A LIFE EXPECTANCY TEST."
290 PRINT "   DO YOU WISH INSTRUCTIONS";
300 INPUT I$
310 IF LEFT$(I$,1)="N" THEN 470
320 PRINT:PRINT
340 PRINT "   THIS IS A TEST TO PREDICT YOUR LIFE EXPECTANCY.  I"
350 PRINT "WILL ASK YOU A SERIES OF SHORT QUESTIONS, WHICH YOU WILL"
360 PRINT "REPLY BY TYPING IN THE CORRESPONDING ANSWER TO THE"
370 PRINT "QUESTION."
380 PRINT
390 PRINT "         EXAMPLE:  WHAT IS YOUR SEX?"
400 PRINT "                   M=MALE"
410 PRINT "                   F=FEMALE"
420 PRINT "'M' AND 'F' ARE THE POSSIBLE REPLIES TO THE QUESTION, ANSWER"
430 PRINT "LIKE THIS:"
440 PRINT "                CHOOSE ONE OF THE LETTERS ABOVE? M"
450 PRINT "TYPING AN 'M' SIGNIFIES YOU ARE A MALE."
460 PRINT:PRINT:PRINT
470 R5=1
480 Z=72
490 A$="ABCDEMGHIJKLFNO"
500 GOTO 1700
510 R5=R5+1
520 IF R5>21 THEN 1900
530 DATA "+++SEX+++"
540 DATA "ARE YOU MALE OR FEMALE?"
550 DATA "M= MALE."
560 DATA " F= FEMALE."
570 DATA 2,"MF"
580 DATA "+++LIFE STYLE+++"
590   DATA "WHERE DO YOU LIVE?"
600 DATA "G= IF YOU LIVE IN AN URBAN AREA WITH A POPULATION OVER 2 MIL."
610 DATA "K= IF YOU LIVE IN A TOWN UNDER 10,000, OR ON A FARM."
620 DATA " I= NEITHER."
630 DATA 3,"GKI"
640 DATA "HOW DO YOU WORK?"
650 DATA "M= IF YOU WORK BEHIND A DESK."
660 DATA "L= IF YOUR WORK REQUIRES HEAVY PHYSICAL LABOR."
670 DATA " I= NONE OF THE ABOVE."
680 DATA 3,"MLI"
690 DATA "HOW LONG DO YOU EXERCISE STRENUOUSLY,"
700 DATA "(TENNIS, RUNNING, SWIMMING, ETC.)?"
710 DATA "F= FIVE TIMES A WEEK FOR AR LEAST A HALF HOUR."
720 DATA "K= JUST TWO OR THREE TIMES A WEEK."
730 DATA " I= DO NOT EXERCISE IN THIS FASHION."
740 DATA 3,"FKI"
750 DATA "WHO DO YOU LIVE WITH?"
760 DATA "M= IF YOU LIVE WITH A SPOUSE, FRIEND, OR IN A FAMILY."
770 DATA "H= IF YOU'VE LIVED ALONE FOR 1-10 YEARS SINCE AGE 25."
780 DATA "G= FOR 11-20 YEARS."
790 DATA "M= FOR 21-30 YEARS."
800 DATA "E= FOR 31-40 YEARS."
810 DATA " M= MORE THAN 40 YEARS."
820 DATA 6,"MHGMED"
830 DATA "+++PSYCHE+++"
840 DATA "DO YOU SLEEP MORE THAN 10 HOURS A NIGHT?"
850 DATA "I= NO."
860 DATA " E=YES."
870 DATA 2,"IE"
880 DATA "+++MENTAL STATE+++"
890 DATA "M= IF YOU ARE INTENSE, AGGRESSIVE, OR EASILY ANGERED."
900 DATA "L= IF YOU ARE EASY GOING, RELAXED, OR A FOLLOWER."
910 DATA " I= NEITHER."
```

```
920 DATA 3,"MLI"
930 DATA "+++HOW YOU FEEL+++"
940 DATA "ARE YOU HAPPY OR UNHAPPY?"
950 DATA "J= HAPPY."
960 DATA "G= UNHAPPY."
970 DATA " I= NEITHER."
980 DATA 3,"JGI"
990 DATA "+++FACTORS+++"
1000 DATA "HAVE YOU HAD A SPEEDING TICKET IN THE LAST YEAR?"
1010 DATA "H= YES."
1020 DATA " I=NO."
1030 DATA 2,"HI"
1040 DATA "+++INCOME+++"
1050 DATA "DO YOU EARN MORE THAN $50,000 A YEAR?"
1060 DATA "G= YES."
1070 DATA " I=NO."
1080 DATA 2,"GI"
1090 DATA "+++SCHOOLING+++"
1100 DATA "J= IF YOU HAVE FINISHED COLLEGE."
1110 DATA "L= IF YOU HAVE FINISHED COLLEGE WITH A GRADUATE"
1120 DATA "OR PROFESSIONAL DEGREE."
1130 DATA " I= NOTHING LISTED."
1140 DATA 3,"JLI"
1150 DATA  "+++AGE+++"
1160 DATA "ARE YOU 65 OR OLDER AND STILL WORKING?"
1170 DATA "L= YES."
1180 DATA " I= NO."
1190 DATA 2,"LI"
1200 DATA "+++HEREDITY+++"
1210 DATA "K= IF ANY GRANDPARENTS LIVED TO 85 YEARS OLD."
1220 DATA "O= IF ALL FOUR GRANDPARENTS LIVED TO 80 YEARS OLD."
1230 DATA " I= NO GRANDPARENTS QUALIFY IN THE ABOVE."
1240 DATA 3,"KOI"
1250 DATA "HAS ANY PARENT DIED OF A STROKE OR HEART ATTACK"
1260 DATA "BEFORE THE AGE OF 50?"
1270 DATA "E= YES."
1280 DATA " I= NO."
1290 DATA 2,"EI"
1300 DATA "+++FAMILY DISEASES+++"
1310 DATA "ANY PARENT, BROTHER, OR SISTER UNDER 50 HAS (OR HAD) "
1320 DATA "CANCER, A HEART CONDITION, OR DIABETES SINCE CHILDHOOD?"
1330 DATA "M= YES."
1340 DATA " I= NO."
1350 DATA 2,"MI"
1360 DATA "+++HEALTH+++"
1365 DATA "HOW MUCH DO YOU SMOKE?"
1370 DATA "A= IF YOU SMOKE MORE THAN TWO PACKS A DAY."
1380 DATA "C= ONE TO TWO PACKS A DAY."
1390 DATA "M= ONE HALF TO ONE PACK A DAY."
1400 DATA " I= DON'T SMOKE."
1410 DATA 4,"ACMI"
1420 DATA "+++DRINK+++"
1430 DATA "DO YOU DRINK THE EQUIVALENT OF A "
1440 DATA "QUARTER BOTTLE OF ALCOHOLIC BEVERAGE A DAY?"
1450 DATA "M= YES."
1460 DATA " I= NO."
1470 DATA 2,"MI"
1480 DATA "+++WEIGHT+++"
1490 DATA "A= IF YOU ARE OVERWEIGHT BY 50 POUNDS OR MORE."
1500 DATA "E= OVER BY 30-50 POUNDS."
1510 DATA "O= OVER BY 10-30 POUNDS."
1520 DATA " I= NOT OVER WEIGHT."
1530 DATA 4,"AEGI"
1540 DATA "+++CHECKUPS+++"
1550 DATA "DO YOU?  IF YOU ARE A MALE OVER 40 HAVE AN ANNUAL CHECKUP?"
1560 DATA "K= YES."
1570 DATA " I= IF NO OR NOT A MALE OR UNDER 40 YEARS OLD."
1580 DATA 2,"KI"
1590 DATA "DO YOU? IF YOU ARE A WOMAN SEE A GYNECOLOGIST ONCE A YEAR?"
1600 DATA "K= YES."
1610 DATA " I= IF NO OR NOT A WOMAN."
1620 DATA 2,"KI"
1630 DATA "+++CURRENT AGE+++"
1640 DATA "K= IF YOU ARE BETWEEN 30 AND 40 YEARS OLD."
1650 DATA "L= BETWEEN 40 AND 50."
1660 DATA "F= BETWEEN 50 AND 70."
1670 DATA "N= OVER 70."
1680 DATA " I= UNDER 30."
1690 DATA 5,"KLFNI"
1700 FOR Q=1 TO 7
1710 READ Q$
1720 IF LEFT$(Q$,1)=" " THEN 1750
1730 PRINT "  ";Q$
1740 NEXT Q
1750 PRINT " ";Q$
1760 READ C,C$
1770 PRINT "CHOOSE ONE OF THE LETTERS ABOVE";
1780 INPUT G$
1790 FOR C2=1 TO C
1800 IF LEFT$(G$,1)= MID$(C$,C2,1) THEN 1830
1810 NEXT C2
1820 GOTO 1770
1830 PRINT
1840 FOR N=1 TO 15
1850 IF LEFT$(G$,1)=MID$(A$,N,1) THEN 1870
1860 NEXT N
1870 M=N-9
1880 Z=Z+M
1890 GOTO 510
1900 PRINT  "YOU ARE EXPECTED TO LIVE TO THE AGE OF";Z;"YEARS"
1910 IF Z<60 THEN 1980
1920 FOR Y=60 TO Z STEP 5
1930 READ M$,F$
1940 NEXT Y
1950 DATA "26%","15%","36%","20%","48%","30%","61%","39%"
1960 DATA "75%","53%","87%","70%","96%","88%","99.9%","99.6%"
1970 PRINT "OUT LIVING ";M$;" OF THE MEN AND ";F$;" OF THE WOMEN."
1980 END
Ok
```

Building a Mouse

Jerry W. Lee



Does The Name Pavlov Ring A Bell?

The theory of learning has been central to experimental psychology since Pavlov's dogged experiments over fifty years ago. Some theorists (like Hull and, more recently, Skinner) think learning theory covers all behavior, and that such areas of psychology as perception, personality and human motivation can eventually be predicted and controlled within its framework.

To the learning theorist, every organism is always in a "learning" situation; but starkly simplified learning experiments, like running mice through mazes, are seen as the clearest way to experiment in this field. Learning takes place when reward (or "reinforcement") occurs; under reinforcement, the organism becomes more likely to emit a certain "behavior." Punishment, or negative reinforcement, doesn't work as well.

Enough said. Here's a simplified mouse you can train all you like. But be aware that while *you* are training *it, it* is training *you.* —TN

Most people would be happy to get rid of mice. Our problem was a lack of them — or rather the space, supplies, and time necessary to raise them for use in an introductory psychology class. One of the most effective ways of teaching students the basic principles of learning is to give them actual "hands on" experience in training an animal. Mice, or more frequently rats, are used for this purpose because they take up little space, are relatively easy to raise, and learn relatively rapidly. Nevertheless, when one has four or five hundred students a year taking introductory psychology, the cost of raising and caring for enough mice or rats for all of these students is prohibitive. It is a rare psychology department that can afford such an expensive learning aid.

Computers, on the other hand, do not have to be fed or cleaned up after, and can easily be taught. While their initial cost is high, their upkeep is low (compared to that of a rat colony), and they can do many other things besides run mazes or push bars in a Skinner box.

For these reasons I decided to attempt to develop a computer program to simulate a mouse which could then be trained by students using our PDP 11/40, under the RSTS-E time sharing system. The program would be written in DEC Basic -PLUS. I decided to call the simulation a mouse simulation rather than a rat simulation simply because mice have received much better press than rats.

Jerry W. Lee, Department of Psychology, Loma Linda University, Riverside, CA 92515.

The "brain" of a computer is well understood. It has to be, since humans manufacture it. But the brain of a mouse is quite complex, in comparison to a digital computer, and is not at all well understood. No one so far has succeeded in developing a computer program that can simulate all the behavior of a mouse. Any computer program which was developed to simulate a mouse would have to be far more limited in its repertoire of possible behavior.

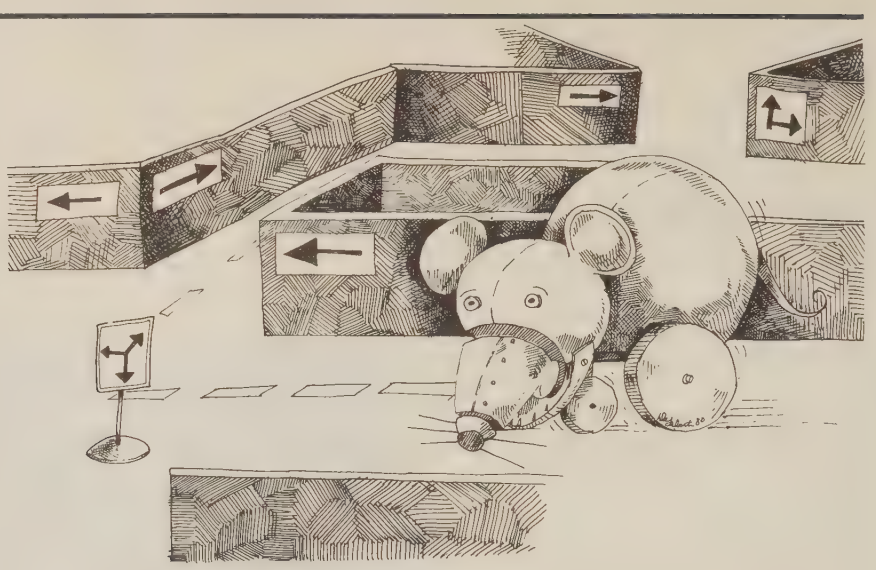
I decided that the mouse simulated would be a very simple one. It would only engage in nine "behaviors." The mouse would be represented by an M on a CRT screen — graphics terminals were not available at the time — and would be able to move up or down, right or left, or on any of the four diagonals. The ninth behavior would be remaining in the same place. This, then, was the original conception of the mouse. As programming continued, other elements were added to the original conception until the current version was developed.

In any training situation there must be a goal or a series of sub-goals. For our mouse, the goal is to receive reinforcement and/or avoid punishment. For our students, the goal is to give the mouse a series of reinforcements and/or punishments until the mouse learns to move to the left and reaches a goal represented by a row of G's along the left side of a CRT screen.

Several constraints were placed on the mouse's behavior. First, the mouse had to react visibly when reinforced or punished by the student. This was necessary in order to reinforce the student's behavior. The visible reaction is accomplished by having a word such as "YUM," "MMM!", or "SLURP" briefly appear on the screen in the place of the mouse whenever the student reinforces the mouse. Words such as "UGH," "OUCH," and "SCREECH" appear when the student punishes the mouse. The student delivers reinforcement to the mouse by pushing the return key on the keyboard. A punishment is delivered whenever the student presses the space bar.

A second constraint on the mouse's behavior was that it had to become more eager to behave when being reinforced, and less eager when being punished. The point of this was to teach students the difficulty of using punishment to train animals to *do* something rather than *not do* something. The problem is that in using punishment to train someone to do something, you must punish all behaviors except the correct behavior. In the process an organism may become reluctant to behave at all, because behavior frequently results in punishment.

A third constraint on the mouse was that the mouse shall have a comparatively short short-term memory. One of the clearest findings in the learning literature is that the sooner



reinforcement for punishment occurs after a behavior is emitted the more quickly the behavior is learned. Thus our mouse was constructed so that if a student did not reinforce or punish the mouse promptly (within one second) when it behaves, the behavior reinforced or punished is sitting in one place. Thus, delays in reinforcement result in the mouse learning to sit still. Delays in punishment result in the mouse learning to move, but learning to move in a random manner.

The Learning Algorithm

Since this mouse was to be a simple one, a simple learning algorithm was devised. A simplified version of the overall algorithm is presented in Figure 1 and Figure 2. (Since this flowchart is intended to clarify the method by which the mouse learns, it does not include the routines that implement the first and second constraints discussed above, or the provision for a "smart" or "dumb" mouse, or the end routine.) For information about these routines you may consult the program itself, which accompanies this article.

The mouse's learning is handled in the following way: A set of sixty memory locations in the computer are randomly assigned the numbers one through nine. These numbers represent the mouses' nine possible behaviors. The computer randomly selects one of the sixty memory locations. If the behavior represented by the mouse does not move the mouse off the CRT screen, then the computer mouse engages in the indicated behavior. If the mouse reaches the goal the training trial is terminated. If reinforcement occurs within one second, the mouse enters the reinforcement subroutine. If punishment occurs, the mouse enters the punishment subroutine. These subroutines will be discussed later. If reinforcement occurs after one second has elapsed, but prior to the expiration of a short wait before the next behavior, the reinforcement subroutine is entered — but the behavior reinforced is "sitting in the same place." If punishment occurs after one second has elapsed, but prior to the expiration of a short wait before the next behavior, then the punishment subroutine is entered — but the behavior punished is "sitting in the same place." If no reinforcement or punishment occurs prior to the expiration of the second wait, then the extinction subroutine is entered. Entry into the punishment or reinforcement subroutine causes the program to enter another wait of variable length (W1% seconds). If reinforcement or punishment occurs during this wait, the behavior reinforced or punished is "sitting," and the wait begins again. If no reinforcement or punishment occurs, then a new memory location from the sixty-element array is selected and the sequence begins again. This last wait, and the reinforcement or punishment of sitting, are necessary to prevent over-eager students from speeding up the mouse by simply terminating each random wait period by delivering reinforcers or punishers.

It may be helpful, at this point, to explain the concept of a wait. WAIT is a statement in DEC Basic-PLUS which causes the computer to wait for input a specified number of seconds. If no input occurs before the specified number of seconds expires, then a trappable error occurs. An error subroutine can then be used to tell the computer what to do in that case.

The reinforcement subroutine. The steps that take place when this subroutine is entered are as follows. The sixty-element memory array is searched *n* times. The value for *n* depends on the smartness-level of the mouse selected by the students. The object of each search is to find numbers representing behaviors which are *not* similar to or identical with the behavior just engaged in. Suppose that the mouse had been reinforced for moving directly to the left. Movements diagonally upward and to the left or diagonally downward and to the left would be similar to the reinforced behavior. Once the routine finds a number representing a behavior not similar to the direction that has just been reinforced, then this number is replaced with the number representing the reinforced behavior. Thus the sixty-element array becomes increasingly loaded with behaviors similar to those which have been reinforced. Since behaviors similar to the reinforced behavior are not replaced by this method, the result also weights the array to simulate some generalization of response. The program then enters the wait-state described earlier.

In addition, the maximum wait of the mouse before the next behavior occurs is decreased if the minimum wait has not been reached. This causes the mouse to appear more "eager" to behave.

The punishment subroutine. The action of the punishment subroutine is very similar to the action of the reinforcement routine. The computer makes *n* searches through the sixty-element array until a number representing a behavior identical with the behavior punished is found. This number is then replaced with a number which is *not* similar to or identical with

```
5 !
!
!       M O U S E    --    A    S I M U L A T I O N
!
!       F O R      G E N E R A L      P S Y C H O L O G Y
!
!              COPYRIGHT 1979 BY JERRY W. LEE

10      RANDOMIZE

15      ON ERROR GO TO 810

20      DIM M%(61%),R$(6%),P$(6%),B$(9%)
        ! M%() = MOUSE'S MEMORY.
        ! R$() = REINFORCEMENT WORDS.
        ! P$() = PUNISHMENT WORDS.
        ! B$() = BEHAVIORS IN THE FORM OF CURSOR MOVEMENT COMMANDS.

30      DATA " OUCH! "," UGH "," EEEK! "," GRUNT ","SCREECH","@*$#%&!"

50      READ P$(I%) FOR I%=1% TO 6%

60      DEF FNR1%=9%*RND(0)+1%
        ! RANDOM NUMBER REPRESENTING ONE OF NINE BEHAVIORS.

64        R2% = 2%
        \ P2% = 4%
        \ W3% = 8%
        \ N1% = 60%
        \ W4% = 80%
        \ G%  = 12%
        \ W%  = 1%
        \ M3% = 5%
        ! REINFORCEMENT SMARTNESS (LARGER NUMBER MAKES SMARTER).
        ! PUNISHMENT SMARTNESS.
        ! MAXIMUM WAIT TO SIT (IN SEC.).
        ! SIZE OF MOUSE'S MEMORY
        ! WIDTH OF CRT SCREEN IN CHARACTERS.
        ! NUMBER OF LINES ON CRT SCREEN
        !AMOUNT TO WAIT FOR REINF. OR PUNISH. BEFORE TRANSFERING TO
                SIT ROUTINE.
        ! SIZE OF LEFT OR RIGHT MOVE.

66      R2%=2% IF R2%<2%
        \ P2%=2% IF P2%<2%

80      DEF FNR2%=N1%*RND(0)+1%
        ! OBTAINS RANDOM NUMBER TO SELECT MEMORY LOCATION IN M%().

90      DATA"SMACK","YUM! ","CHOMP","BURP!","MMM!!","SLURP"

110     READ R$(I) FOR I=1 TO 6

120     PRINT CHR$(155%)+'H'+CHR$(155%)+'J'
        ! HOME AND CLEAR SCREEN.

130     PRINT"THIS PROGRAM IS DESIGNED TO SIMULATE";
        \ PRINT" THE OPERANT CONDITIONING OF A MOUSE."
        \ PRINT"THE OBJECT IS TO GET THE MOUSE TO";
        \ PRINT" GO TO THE GOAL BY REINFORCING THE CORRECT"

135     PRINT"RESPONSES AND/OR PUNISHING INCORRECT RESPONSES.";
        \ PRINT"  THE MOUSE IS REPRESENTED BY"
        \ PRINT"THE LETTER M.  THE GOAL IS REPRESENTED BY A LINE OF G'S."

140     PRINT"TO REINFORCE THE MOUSE PUSH THE RETURN ";
        \ PRINT"KEY WITHIN ONE SECOND AFTER HE MOVES."
        \ PRINT"TO PUNISH HIM PUSH THE SPACE BAR.  IF YOU TAKE LONGER";
        \ PRINT " THAN A SECOND YOU MAY"

145     PRINT"CONFUSE THE MOUSE."

150     L$=CHR$(8%)
        \ R$=CHR$(155%)+'C'
        \ U$=CHR$(155%)+'A'
        \ D$=CHR$(10%)
        \ B$=CHR$(13%)
        \ H$=CHR$(155%)+'H'
        \ C$=CHR$(155%)+'J'
        \ E$=CHR$(155%)+'K'
        ! ** CURSOR CONTROL FUNCTIONS **
        ! L$ = LEFT.
        ! R$ = RIGHT.
        ! U$ = UP.
        ! D$ = DOWN.
        ! H$ = HOME.
        ! C$ = CLEAR TO END OF SCREEN.
        ! E$ = ERASE TO END OF LINE.

210     S$=L$+" "+L$
        ! IF PRINTED ERASES MOUSE AND PLACES CURSOR IN MOUSE'S FORMER
                POSITION.

230     B$(4%)=S$
        \ B$(8%)=S$

240     FOR I%=1% TO M3%
        \ B$(4%)=B$(4%)+R$
        \ B$(8%)=B$(8%)+L$
        \ NEXT I%
        ! LOOP PUTS IN RIGHT OR LEFT COMPONENT OF BEHAVIOR.

250     B$(1%)=B$(8%)+U$
        \ B$(2%)=S$+U$
        \ B$(3%)=B$(4%)+U$
        \ B$(5%)=B$(4%)+D$
        \ B$(6%)=S$+D$
        \ B$(7%)=B$(8%)+D$
        \ B$(9%)=S$+S$+S$+S$+S$
        \ B$(9%)=B$(9%)+"    "
        ! INSTALL BEHAVIORS.
        ! 1 = LEFT & UP.
```

```
        ! 2 = UP.
        ! 3 = RIGHT & UP.
        ! 4 = RIGHT.
        ! 5 = RIGHT & DOWN.
        ! 6 = DOWN.
        ! 7 = LEFT & DOWN.
        ! 8 = LEFT.
        ! 9 = REMAIN IN SAME PLACE.

320     OPEN "KB: " AS FILE 1%

340     M%(I%)=FNR1% FOR I%=1% TO N1%
        \ W1%=(W3%-1%)/2%
        \ W1%=1% IF W1%<=0%
        ! RANDOMIZE MOUSE'S MEMORY
        ! SET VALUE OF STANDARD WAIT TO HALF MAXIMUM.

350     PRINT
        \ INPUT"DO YOU WANT A SMART MOUSE OR A DUMB MOUSE";S$

360     IF S$="S"THEN S9%=-1%
        \ GO TO 390
        ! S9% = SMART FLAG = -1 FOR SMART MOUSE = 0 OTHERWISE.

370     IF S$="D" THEN S9%=0%
        \ GO TO 390

380     PRINT"TYPE AN S OR A D PLEASE."
        \ GO TO 350

390     PRINT
        \ M% = 0%
        \ R% = 0%
        \ P1%= 0%
        \ INPUT"PUSH THE RETURN BUTTON TO START THE GAME";Q$
        ! PRINT A LINE.
        ! ZERO MOVE COUNTER.
        ! ZERO REINFORCEMENT COUNTER.
        ! ZERO PUNISHMENT COUNTER.
        ! START GAME.

400     PRINT H$+C$
        ! HOME & CLEAR.

410     !

        B E G I N    S I M U L A T I O N

430     PRINT 'G' FOR I%=1% UNTIL I%>=G%-1%
        ! PRINT GOAL.

450     D1$=''
        \ D1$=D1$+D$ FOR I%=1% TO G%/2%
        \ PRINT H$;D1$;'G';TAB(25%);
        ! MOVE CURSOR TO APPROX. MIDDLE OF SCREEN.

460     S$=SYS(CHR$(3%))
        ! TURN OFF TERMINAL ECHOING.

470     PRINT "LET'S START.  HERE'S THE MOUSE."
        ! STARTING SIGN.

480     SLEEP 3%

490     PRINT U$;B$;R$;E$;
        ! ERASE STARTING SIGN.

500     PRINT B$;
        \ PRINT R$;TAB(60%);
        ! MOVE CURSOR TO MOUSE'S STARTING POSITION.

520     T1=TIME(0%)
        \ H%=60%
        \ V%=G%/2%+1%
        ! T1 = STARTING TIME.
        ! H% = HORIZONTAL POSITION.
        ! V% = VERTICAL POSITION.

540     PRINT "M";

550     GOSUB 1300
        ! SIT FOR AWHILE BEFORE BEHAVING.

555     N2%=FNR2%
        \ N%=M%(N2%)
        \ R1% = 0%
        ! SELECT BEHAVIOR TO TEST
        ! THEN ZERO RESPONSE FLAG (R%)

570     I1%=0
        ! ZERO ILLEGAL BEHAVIOR COUNTER.

590     I1%=I1%+1%
        \ IF I1%>20% THEN M%(N2%)=FNR1%
        \ N%=M%(N2%)
        ! IF MORE THAN TWENTY ILLEGAL BEHAVIORS ARE ARE ATTEMPTED TAKE A
                RANDOM MOVE.
        ! AN ILLEGAL BEHAVIOR ONE WHICH WOULD TAKE THE MOUSE OFF THE SCREEN.

600     ON N% GO TO 610, 620, 630, 640, 650, 660, 670, 680, 720
        ! CHECK FOR LEGALITY OF BEHAVIOR.

610     H1%=-M3%
        \ V1%=-1%
        \ GO TO 690

620     H1%=0%
        \ V1%=-1%
        \ GO TO 690

630     H1%=M3%
        \ V1%=-1%
        \ GO TO 690

640     H1%=M3%
        \ V1%=0%
        \ GO TO 690

650     H1%=M3%
        \ V1%=+1%
        \ GO TO 690

660     H1%=0%
        \ V1%=+1%
        \ GO TO 690

670     H1%=-M3%
        \ V1%=+1%
        \ GO TO 690

680     H1%=-M3%
        \ V1%=0%
        \ GO TO 690
```

```
690     IF H%+H1%<=1% GO TO 850
        ! IF GOAL HAS BEEN REACHED GO TO END ROUTINE.

700     H2%=H%+H1%
        \ V2%=V%+V1%
        \ IF H2%>=W4% OR V2%>=G% OR V2%<1% THEN N2%=FNR2%
        \ N%=M%(N2%)
        \ GO TO 590
        ! IF SELECTED BEHAVIOR IS ILLEGAL (MOVES MOUSE OFF SCREEN) THEN
                SELECT A NEW BEHAVIOR AND GO TO THE BEGINING OF
                THE TEST ROUTINE.

710     H%=H2%
        \ V%=V2%
        \ M% = M% + 1%
        ! RECORD NEW HORIZONTAL POSITION.
        ! RECORD NEW VERTICAL POSITION.
        ! COUNT MOVE.

720     PRINT B$(N%);"M";
        ! MOUSE BEHAVES !!!

740     WAIT W%
        ! GIVE USER W% SECONDS TO RESPOND TO MOVE.
        ! EXHAUSTED WAIT GOES TO ERROR ROUTINE WHICH RESUMES AT SIT ROUTINE.

770     S$=SYS(CHR$(4%))
        \ S$=SYS(CHR$(11%))
        ! SYS(CHR$(4%)) = SET SINGLE CHARACTER GET MODE.
        ! SYS(CHR$(11%))= CLEAR KEYBOARD BUFFER.

780     FIELD #1%, 10% AS R1$
        \ GET #1%
        \ R1$=LEFT(R1$,1%)
        \ R1%=-1%
        ! GET USER'S RESPONSE.
        ! SET RESPONSE FLAG (R%) TO INDICATE A RESPONSE WAS MADE.

790     IF R1$=CHR$(32%) THEN GOSUB 1030 ELSE GOSUB 1150
        ! IF SPACE BAR HIT GO TO PUNISHMENT ROUTINE OTHERWISE GO TO
                REINFORCEMENT ROUTINE.

794     GOSUB 1300
        ! GO TO SIT SUBROUTINE.
        ! PROGRAM ALSO TRANSFERS TO THIS LINE WHEN WAIT EXHAUSTED ERROR
                OCCURS IN LINE 780.

796     GOSUB 817 IF NOT R1%
        ! GO TO EXTINCTION ROUTINE IF NO RESPONSE FROM USER IN LINE 780.

800     GO TO 555
        ! RETURN TO SELECT ANOTHER BEHAVIOR.

805     !

        ! E R R O R    H A N D L I N G    R O U T I N E S

810     IF ERR<>15 THEN ON ERROR GO TO 0
        ! IF ERROR IS NOT AN EXHAUSTED WAIT THEN TERMINATE PROGRAM.

811     IF ERL=780 THEN RESUME 794
        ! USER DID NOT RESPOND TO BEHAVIOR WITHIN W% SECONDS.

815     IF ERL=1320 THEN RESUME 1350
        ! USER DID NOT RESPOND DURING SIT PERIOD.

816     ON ERROR GO TO 0
        ! AN UNPLANNED ERROR HAS OCCURED, TERMINATE THE PROGRAM.

817     !

                E X T I N C T I O N    R O U T I N E

820     M%(N2%)=FNR1%
        ! REPLACE LAST USED BEHAVIOR IN MEMORY WITH A RANDOM BEHAVIOR.

840     RETURN
        ! RETURN TO MAIN PROGRAM FOR RETURN
        ! BEHAVIORAL SELECTION ROUTINE.

850     !

                E X I T    R O U T I N E

860     S$=SYS(CHR$(2%))
        ! ENABLE TERMINAL ECHOING

870     PRINT B$(N%);'M';B$;
        \ V%=V%+V1%
        \ M%=M%+1%
        ! MOVE TO GOAL!!!
        ! INCREMENT MOVEMENT COUNTER.

880     FOR I%=1% TO 8%
        \ PRINT H$;
        \ FOR J%=1% TO G%-1%
        \ PRINT D$;' ';L$;
        \ NEXT J%
        ! START GOAL FLASHING LOOP
        ! BLANK OUT GOAL

920     PRINT H$;
        \ FOR J%=1% TO G%-1%
        \ PRINT D$;'G';L$;
        \ NEXT J%
        \ NEXT I%
        ! PRINT GOAL AGAIN
        ! RETURN TO LOOP START AND CONTINUE TO FLASH GOAL 8 TIMES.
950     PRINT H$+C$+D$;TAB(20%);"CONGRATULATIONS"

960     PRINT "THE MOUSE MADE IT TO THE GOAL IN";TIME(0%)-T1;"SECONDS."

970     PRINT "YOU HAVE USED";R%;"REINFORCEMENTS AND";
        \ PRINT P1%;"PUNISHMENTS FOR THIS MOUSE."

980     PRINT "THIS MOUSE HAS MOVED";M%;"TIMES."

990     WAIT 0%
        ! CANCEL WAIT STATE.

1000    INPUT"DO YOU WANT TO TRY AGAIN";Q$
        \ IF LEFT(Q$,1%)="Y" THEN PRINT
        \ PRINT"DO YOU WANT TO USE YOUR OLD MOUSE (TYPE O) ";
        \ INPUT"OF DO YOU WANT A NEW MOUSE (TYPE N)";Q$
        \ IF LEFT(Q$,1%)="N" GO TO 340 ELSE GO TO 390

1005    IF LEFT(Q$,1%)<>"N" THEN PRINT"ANSWER YES OR NO PLEASE."
        \ GO TO 1000

1010    PRINT H$;C$
        ! HOME CURSOR AND CLEAR SCREEN.
```

```
1020    GO TO 4000
        ! GO TO END OF PROGRAM.

1030    !

        ! P U N I S H M E N T    S U B R O U T I N E

1040    PRINT L$;L$;L$;L$;P$(6%*RND(0)+1%);L$;L$;L$;L$;L$;L$;L$;E$;"   M";
        \ P1%=P1%+1%
        ! PRINT RANDOMLY SELECTED PUNISHMENT WORD IN MOUSE'S POSITION
                THEN ERASE AND PRINT MOUSE.
        ! INCREMENT PUNISHMENT COUNTER.

1050    IF S9% THEN J%=P2% ELSE J%=P2%/2%
        ! SET FOLLOWING LOOP TO REPEAT FEWER TIMES FOR DUMB MOUSE.

1060    FOR I%=1% TO J%
        \ FOR K%=1% UNTIL N%=M%(K%)OR K%>N1%
        \ NEXT K%
        ! START LOOP.
        ! FIND BEHAVIOR IN ARRAY EQUAL TO PUNISHED BEHAVIOR

1075    GO TO 1110 IF K%>N1%
        ! IF NO SUCH BEHAVIOR END SEARCH LOOP.

1077    IF N%=9% THEN U%=9%
        \ L%=9%
        \ GO TO 1100
        ! IF BEHAVIOR WAS SITTING SET U% AND L% TO SITTING BEHAVIOR
                NUMBER.

1080    P%=M%(K%)
        \ U%=P%+1%
        \ L%=P%-1%
        \ L%=8% IF L%=0%
        \ IF U%=9% THEN U%=1%
        ! SET U% TO BEHAVIOR ONE MOVE CLOCKWISE FROM PUNISHED BEHAVIOR.
        ! SET L% TO BEHAVIOR ONE MOVE COUNTER-CLOCKWISE.

1100    M%(K%)=FNR1% FOR J1%=1% UNTIL M%(K%)<>P% AND M%(K%)<>U% AND M%(K%)<>L%
        ! INSERT ANY BEHAVIOR INTO MEMORY LOCATION OF BEHAVIOR THAT WAS
                PUNISHED EXCEPT PUNISHED BEHAVIOR OR BEHAVIOR SIMILAR TO
                PUNISHED BEHAVIOR.

1110    NEXT I%
        ! SEARCH AND REPLACE PUNISHED BEHAVIORS J% TIMES.

1120    W1%=W1%+1% IF W1%<W3%
        ! INCREASE WAIT IN SIT ROUTINE IF NOT ABOVE MAXIMUM.

1140    RETURN

1150    !

        ! R E I N F O R C E M E N T    S U B R O U T I N E

1160    PRINT L$;L$;L$;R$(6*RND(0)+1%);L$;L$;L$;L$;L$;E$;"  M";
        \ R%=R%+1%
        ! PRINT RANDOM REINFORCEMENT MOVE IN PLACE OF MOUSE
                THEN ERASE AND REPRINT MOUSE.
        ! INCREMENT REINFORCEMENT COUNTER.

1180    IF S9% THEN E%=R2% ELSE E%=R2%/2%
        !  SET SEARCH LOOP TO REPEAT FEWER TIMES IF MOUSE IS DUMB.

1185    IF N%=9% THEN U%,D%=9%
        \ GO TO 1210
        ! IF REINFORCED BEHAVIOR WAS SITTING THEN SET U% AND D%
                TO REPRESENT SITTING BEHAVIOR.

1190    U%=N%+1%
        \ L%=N%-1%
        \ L%=8% IF L%=0%
        \ IF U%=9% THEN U%=1%
        ! SET U% TO ONE MOVE CLOCKWISE TO REINFORCED BEHAVIOR.
        ! SET L% TO ONE MOVE COUNTER-CLOCKWISE.

1210    FOR I%=1% TO E%
        \ FOR J%=1% UNTIL(M%(J%)<>N% AND M%(J%)<>U% AND M%(J%)<>L%) OR
        J%>=N1%
        \ NEXT J%
        \ M%(J%)=N%
        \ NEXT I%
        ! SEARCH ARRAY FOR BEHAVIOR NOT EQUAL TO OR SIMILAR TO
                REINFORCED BEHAVIOR AND REPLACE WITH REINFORCED
                BEHAVIOR.

1250    IF W1%>0% THEN W1%=W1%-1%
        ! DECREASE WAIT TIME IN SIT ROUTINE IF WAIT IS NOT ALREADY
                AT MINIMUM.

1270    RETURN

1300    !

        ! S I T    S U B R O U T I N E

1310    W1%=1% IF W1%<1%
        \ W2%=2%*INT(RND(0)+.5)-1%
        \ W2%=W2%+W1%
        \ IF W2%<1% THEN W2%=1%
        ! MAKE SURE WAIT WILL NOT BE SET TO ZERO.
        ! ADD A RANDOM AMOUNT TO THE SIT PERIOD.

1320    WAIT W2%
        \ FIELD#1%,10% AS R1$
        \ S$=SYS(CHR$(4%))
        \ S$=SYS(CHR$(11%))
        \ GET #1%
        ! SET WAIT.
        ! DEFINE HOW MUCH TO GET FROM KEYBOARD.
        ! SET SINGLE CHARACTER GET MODE.
        ! CLEAR KEYBOARD BUFFER.
        ! GET A CHARACTER.
        ! IF NO RESPONSE FROM USER IN W2% SECONDS GOES TO ERROR
                ROUTINE WHICH RESUMES AT 1350

1330    R1$=LEFT(R1$,1%)
        \ N%=9%
        \ IF R1$=CHR$(32%) THEN GOSUB 1030 ELSE GOSUB 1150
        ! IF RESPONSE WAS SPACE CHARACTER GO TO PUNISHMENT ROUTINE
                OTHERWISE GO TO THE REINFORCEMENT ROUTINE.

1340    GO TO 1300
        ! REPEATS SIT ROUTINE IF MOUSE WAS REINFORCED OR PUNISHED
                DURING THE SIT.  THIS LINE IS BYPASSED WHEN EXHAUSTED
                WAIT OCCURS IN LINE 1320.

1350    RETURN

4000    END
Ready
```

the number representing the behavior punished. If the maximum wait has not already been reached, the mouse's wait before the next behavior is also increased, so that the mouse appears more "reluctant" to behave.

The extinction subroutine. The extinction subroutine is very simple in action. A number between one and nine (representing one of the nine behaviors) is randomly selected and this number is inserted into the memory location which held the most recent behavior engaged in by the mouse. If no reinforcement or punishment occurs for a number of trials, the mouse will gradually "forget" what it has learned.

Use of the Program

The mouse is usually used as an outside assignment for students. Since the program will run in a time-sharing environment, many students can run the program simultaneously. Students are given a handout describing how to operate the computer to run the mouse simulation. They are then asked to train several mice with punishment, several with reward, and to record the amounts of time, and numbers of trials, reinforcements, and punishments needed to train each mouse. They are then asked to read the text material on learning, to describe the differences they found under conditions of reinforcement and punishment, and to discuss why these differences exist and what sort of learning is actually occurring.

As in real life, the performance of the mouse is almost invariably worse with punishment than with reward. The three most common (correct) reasons that students give for this fact are:

1. Punishing the mouse's movement in the wrong direction also punishes movement in general so the mouse becomes less willing to move.
2. Students tend only to punish behaviors which move the mouse away from the goal; irrelevant movements, such as moving up or down, are not punished. Thus it takes longer for the mouse to learn with punishment because it engages in more irrelevant behavior.
3. When the mouse is moving toward the goal nothing happens to it. Thus, the mouse will gradually forget what it has learned. Extinction sets in.

Students generally report that they find the computer experience enjoyable. MOUSE has been in use for approximately 18 months and is believed to be relatively bug-free.

Limitations

As noted earlier, MOUSE is a rather limited mouse. It will behave like a real mouse only when reinforcement occurs for every correct behavior (continuous ratio reinforcement) or punishment occurs for every incorrect behavior. If more complex reinforcement is attempted — for example, reinforcing every third correct behavior (fixed ratio three), then MOUSE's behavior will not duplicate that of a real mouse. In addition, MOUSE's learning curve for continuous-ratio reinforcement does not exactly duplicate that of a real mouse. These problems are being worked on and a possible solution is being attempted. Nevertheless, the current version of MOUSE does serve a useful function in the general psychology lab, in causing students to think about the differences between reinforcement and punishment. □

Chapter Four

Physics

Projectile Motion

Projectile Motion

The path followed by a projectile is called its trajectory. The trajectory is affected to a large extent by air resistance, which makes an exact analysis of the motion extremely complex. We shall, however, neglect the effects of air resistance and assume the motion takes place in empty space.

In the general case of projectile motion, the body (bullet, rocket, mortar, etc.) is given an initial velocity at some angle θ above (or below) the horizontal.

If V_o represents the initial velocity (muzzle velocity), the horizontal and vertical components are:

$$V_{ox} = V_o \cos\theta \ , \ V_{oy} = V_o \sin\theta$$

Since we are neglecting air resistance, the horizontal velocity component remains constant throughout the motion. At any time, it is:

$$V_x = V_{ox} = V_o \cos\theta = \text{constant} \quad (1)$$

The vertical part of the motion is one of constant downward acceleration due to gravity. It is the same as for a body projected straight upward with an initial velocity $V_o \sin\theta$. At a time "t" after the start, the vertical velocity is:

$$V_y = V_{oy} - gt = V_o \sin\theta - gt \quad (2)$$

where "g" is the acceleration due to gravity.

The horizontal distance is given by:

$$x = V_{ox} t = (V_o \cos\theta)\, t \quad (3)$$

and the vertical distance by:

$$y = V_{oy} t - 1/2\, gt^2$$
$$= (V_o \sin\theta)\, t - 1/2\, gt^2 \quad (4)$$

The time for the projectile to return to its initial elevation is found from Equation (4) by setting y = 0. This gives

$$t = \frac{2 V_o \sin\theta}{g} \quad (5)$$

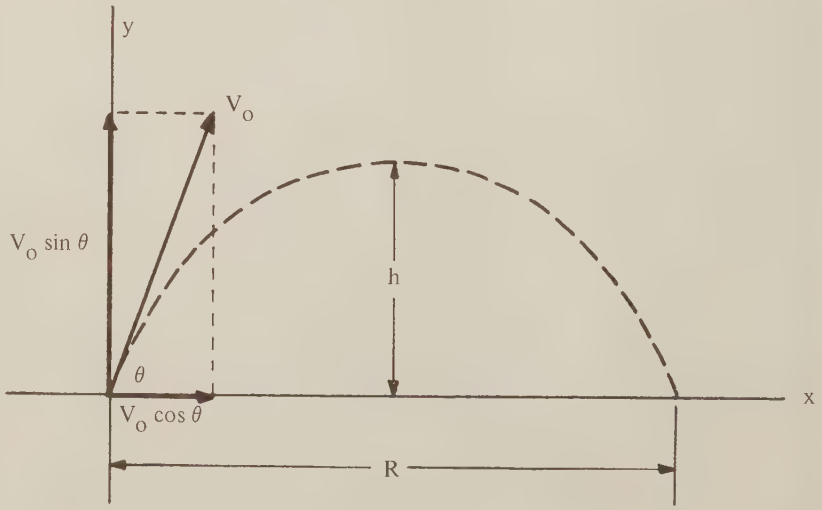
The horizontal distance when the projectile returns to its initial elevation is called the horizontal range, "R." Introducing the time to reach the point in Equation (3), we find:

$$R = \frac{2 V_o \sin\theta \cos\theta}{g} \quad (6)$$

Since $2\sin\theta\cos\theta = \sin 2\theta$, Equation (6) becomes:

$$R = \frac{V_o^2 \sin 2\theta}{g} \quad (7)$$

The horizontal range is thus proportional to the square of the initial



With special acknowledgment to Sears and Zemansky, authors of *University Physics*.

velocity for a given angle of elevation. Since the maximum value of sin 2θ is 1, the maximum horizontal range, Rmax is v_o^2/g. But if sin $2\theta = 1$, then $2\theta = 90°$ and $\theta = 45°$. Hence the maximum horizontal range, in the absence of air resistance, is attained with angle of elevation of 45°.

From the standpoint of gunnery, what one usually wishes to know is what the angle of elevation should be for a given muzzle velocity v_o in order to hit a target whose position is known. Assuming the target and gun are at the same elevation and the target is at a distance R, Equation (7) may be solved for θ.

$$\theta = 1/2 \sin^{-1}\left(\frac{Rg}{V_o^2}\right) = 1/2 \sin^{-1}\left(\frac{R}{R_{max}}\right) \quad (8)$$

Provided R is less than the maximum range, this equation has two solutions for values of θ between 0° and 90°. Either of the angles gives the same range. Of course, time of flight and maximum height reached are both greater for the high angle trajectory.

For example, say the maximum range of our gun is 10,000 yards and the target is at 5,900 yards:

$$\theta = 1/2 \sin^{-1} \frac{5,900}{10,000}$$
$$= 1/2\ 36°$$
$$= 18°, \text{ or } 90° - 18° = 72°$$

Exercise 1

Play the computer game Gunner which appears on the next page. Use trial and error to try and destroy the target (see sample run). You get five chances per target. How many shots did it take to destroy all five targets? Did you ever fail to destroy a target in five trials?

Exercise 2

Now play Gunner but determine the correct firing angle from either sine tables or a slide rule. You should be able to destroy every target with just one shot. What happens when the target is very close? Can you always use whole angles?

Exercise 3

Wire a computer program to accept the maximum range of your gun and the range to the target and then calculate the correct firing angle. You will have to solve two problems to write such a program:

1. The Basic language does not have an ARCSIN function. However, the following formula may help:

$$\sin^{-1} x = \tan^{-1}\left(\frac{x}{\sqrt{1-x2}}\right)$$

2. You must convert from radians to degrees.

Exercise 4

Play the game Guner1 which substitutes a moving target for the stationary one in Gunner. How does your strategy change? Is it useful to use a method other than trial and error to play the game?

Gunner

GUNNER allows you to adjust the fire of a field artillery weapon to hit a stationary target. You specify the number of degrees of elevation of your weapon; 45 degrees provides maximum range with values under or over 45 degrees providing less range.

You get up to five shots to destroy the enemy before he destroys you. Gun range varies between 20,000 and 60,000 yards and burst radius is 100 yards. You must specify elevation within approximately 0.2 degrees to get a hit.

Tom Kloos of the Oregon Museum of Science and Industry in Portland, Oregon originally wrote GUNNER. Extensive modifications were added by David Ahl.

```
10 PRINT TAB(30);"GUNNER"
20 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
30 PRINT:PRINT:PRINT
130 PRINT "YOU ARE THE OFFICER-IN-CHARGE, GIVING ORDERS TO A GUN"
140 PRINT "CREW, TELLING THEM THE DEGREES OF ELEVATION YOU ESTIMATE"
150 PRINT "WILL PLACE A PROJECTILE ON TARGET.  A HIT WITHIN 100 YARDS"
160 PRINT "OF THE TARGET WILL DESTROY IT." : PRINT
170 R=INT(40000*RND(1)+20000)
180 PRINT "MAXIMUM RANGE OF YOUR GUN IS ";R;" YARDS."
185 Z=0
190 PRINT
195 S1=0
200 T=INT(R*(.1+.8*RND(1)))
210 S=0
220 GOTO 370
230 PRINT "MINIMUM ELEVATION IS ONE DEGREE."
240 GOTO 390
250 PRINT "MAXIMUM ELEVATION IS 89 DEGREES."
260 GOTO 390
270 PRINT "OVER TARGET BY";ABS(E);"YARDS."
280 GOTO 390
290 PRINT "SHORT OF TARGET BY"ABS(E);"YARDS."
300 GOTO 390
320 PRINT "*** TARGET DESTROYED ***  ";S;"ROUNDS OF AMMUNITION EXPENDED
325 S1=S1+S
330 IF Z=4 THEN 490
340 Z=Z+1
345 PRINT
350 PRINT "THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY..."
360 GOTO 200
370 PRINT "     DISTANCE TO THE TARGET IS"T;"YARDS."
380 PRINT
390 PRINT
400 INPUT "ELEVATION";B
420 IF B>89 THEN 250
430 IF B<1 THEN 230
440 S=S+1
442 IF S<6 THEN 450
444 PRINT:PRINT "BOOM !!!!   YOU HAVE JUST BEEN DESTROYED ";
446 PRINT "BY THE ENEMY." : PRINT : PRINT : PRINT : GOTO 495
450 B2=2*B/57.3 : I=R*SIN(B2) : X=T-I : E=INT(X)
460 IF ABS(E)<100 THEN 320
470 IF E>100 THEN 290
480 GOTO 270
490 PRINT : PRINT : PRINT "TOTAL ROUNDS EXPENDED WERE:";S1
492 IF S1>18 THEN 495
493 PRINT "NICE SHOOTING !!" : GOTO 500
495 PRINT "BETTER GO BACK TO FORT SILL FOR REFRESHER TRAINING!"
500 PRINT : INPUT "TRY AGAIN (Y OR N)";Z$
510 IF Z$="Y" THEN 170
520 PRINT "OK.  RETURN TO BASE CAMP."
999 END
```

```
                GUNNER
 CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


YOU ARE THE OFFICER-IN-CHARGE, GIVING ORDERS TO A GUN
CREW, TELLING THEM THE DEGREES OF ELEVATION YOU ESTIMATE
WILL PLACE A PROJECTILE ON TARGET.  A HIT WITHIN 100 YARDS
OF THE TARGET WILL DESTROY IT.

MAXIMUM RANGE OF YOUR GUN IS  55684  YARDS.

     DISTANCE TO THE TARGET IS 15755 YARDS.


ELEVATION? 9
OVER TARGET BY 1452 YARDS.

ELEVATION? 8
SHORT OF TARGET BY 407 YARDS.

ELEVATION? 8.2
*** TARGET DESTROYED ***   3 ROUNDS OF AMMUNITION EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY...
     DISTANCE TO THE TARGET IS 11349 YARDS.


ELEVATION? 84
OVER TARGET BY 241 YARDS.

ELEVATION? 84.1
*** TARGET DESTROYED ***   2 ROUNDS OF AMMUNITION EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY...
     DISTANCE TO THE TARGET IS 19146 YARDS.


ELEVATION? 11
OVER TARGET BY 1713 YARDS.

ELEVATION? 10
SHORT OF TARGET BY 102 YARDS.

ELEVATION? 10.06
*** TARGET DESTROYED ***   3 ROUNDS OF AMMUNITION EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY...
     DISTANCE TO THE TARGET IS 10792 YARDS.


ELEVATION? 84.3
OVER TARGET BY 227 YARDS.

ELEVATION? 84.4
*** TARGET DESTROYED ***   2 ROUNDS OF AMMUNITION EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY...
     DISTANCE TO THE TARGET IS 36976 YARDS.


ELEVATION? 21
OVER TARGET BY 282 YARDS.

ELEVATION? 20.8
*** TARGET DESTROYED ***   2 ROUNDS OF AMMUNITION EXPENDED


TOTAL ROUNDS EXPENDED WERE: 12
NICE SHOOTING !!

TRY AGAIN (Y OR N)? N
OK.  RETURN TO BASE CAMP.
```

Kinema

This program tests your fundamental knowledge of kinematics. It presents a simple problem: a ball is thrown straight up in the air at some random velocity. You then must answer three questions about the flight of the ball:

1. How high will it go?
2. How long until it returns to earth?
3. What will be its velocity after a random number of seconds?

The computer evaluates your performance; within 15% of the correct answer is considered close enough. After each run, the computer gives you another problem until you interrupt it.

KINEMA was shortened from the original Huntington Computer Project Program, KINERV, by Richard Pav of Patchogue High School, Patchogue, New York.

```
                KINEMA
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY


A BALL IS THROWN UPWARDS AT 35 METERS PER SECOND.

HOW HIGH WILL IT GO (IN METERS)? 10
NOT EVEN CLOSE....
CORRECT ANSWER IS  61.25

HOW LONG UNTIL IT RETURNS (IN SECONDS)? 7
CLOSE ENOUGH.
CORRECT ANSWER IS  7

WHAT WILL ITS VELOCITY BE AFTER 4.5 SECONDS? 20
NOT EVEN CLOSE....
CORRECT ANSWER IS -10


 1 RIGHT OUT OF 3.

A BALL IS THROWN UPWARDS AT 25 METERS PER SECOND.

HOW HIGH WILL IT GO (IN METERS)? 45
NOT EVEN CLOSE....
CORRECT ANSWER IS  31.25

HOW LONG UNTIL IT RETURNS (IN SECONDS)? 4
NOT EVEN CLOSE....
CORRECT ANSWER IS  5

WHAT WILL ITS VELOCITY BE AFTER 3.2 SECONDS? 12
NOT EVEN CLOSE....
CORRECT ANSWER IS -7


 0 RIGHT OUT OF 3.
```

```
10 PRINT TAB(33);"KINEMA"
20 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
30 PRINT: PRINT: PRINT
100 PRINT
105 PRINT
106 Q=0
110 V=5+INT(35*RND(1))
111 PRINT "A BALL IS THROWN UPWARDS AT";V;"METERS PER SECOND."
112 PRINT
115 A=.05*V^2
116 PRINT "HOW HIGH WILL IT GO (IN METERS)";
117 GOSUB 500
120 A=V/5
122 PRINT "HOW LONG UNTIL IT RETURNS (IN SECONDS)";
124 GOSUB 500
130 T=1+INT(2*V*RND(1))/10
132 A=V-10*T
134 PRINT "WHAT WILL ITS VELOCITY BE AFTER";T;"SECONDS";
136 GOSUB 500
140 PRINT
150 PRINT Q;"RIGHT OUT OF 3.";
160 IF Q<2 THEN 100
170 PRINT "  NOT BAD."
180 GOTO 100
500 INPUT G
502 IF ABS((G-A)/A)<.15 THEN 510
504 PRINT "NOT EVEN CLOSE...."
506 GOTO 512
510 PRINT "CLOSE ENOUGH."
511 Q=Q+1
512 PRINT "CORRECT ANSWER IS ";A
520 PRINT
530 RETURN
999 END
```

Lunar

This game in its many different versions and names (ROCKET, LUNAR, LEM, and APOLLO) is by far and away the single most popular computer game. It exists in versions that start you anywhere from 500 feet to 200 miles above the moon, or other planets, too. Some allow the control of directional stabilization rockets and/or the retro rocket. The three versions presented here represent the most popular of the many variations.

In most versions of this game, the temptation is to slow up too soon and then have no fuel left for the lower part of the journey. This, of course, is disasterous (as you will find out when you land your own capsule)!

LUNAR was originally in FOCAL by Jim Storer while a student at Lexington High School and subsequently converted to BASIC by David Ahl.

In this program, you set the burn rate of the retro rockets (pounds of fuel per second) every 10 seconds and attempt to achieve a soft landing on the moon. 200 lbs/sec really puts the brakes on, and 0 lbs/sec is free fall. Ignition occurs at 8 lbs/sec, so *do not* use burn rates between 1 and 7 lbs/sec. To make the landing more of a challenge, but more closely approximate the real Apollo LEM capsule, you should make the available fuel at the start (N) equal to 16,000 lbs, and the weight of the capsule (M) equal to 32,500 lbs in Statement 15.

Some versions of BASIC object to the series expansion calculations in Statements 420 and 430 (as you near the lunar surface, these numbers get very small). If your does, substitute the following expanded form for the expansion in Statement 420:

```
-Q*(1+Q*(1/2+Q*(1/3+Q*(1/4+Q/5))))
```

You should be able to figure the other one out yourself.

```
                                LUNAR
 CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

 THIS IS A COMPUTER SIMULATION OF AN APOLLO LUNAR
 LANDING CAPSULE.

 THE ON-BOARD COMPUTER HAS FAILED
 SO YOU HAVE TO LAND THE CAPSULE MANUALLY.

 SET BURN RATE OF RETRO ROCKETS TO ANY VALUE BETWEEN
 0 (FREE FALL) AND 200 (MAXIMUM BURN) POUNDS PER SECOND.
 SET NEW BURN RATE EVERY 10 SECONDS.

 CAPSULE WEIGHT 32,500 LBS; FUEL WEIGHT 16,500 LBS.


GOOD LUCK

SEC        MI + FT       MPH          LB FUEL       BURN RATE

 0          120  0        3600         16500         ? 0
 10         109  5015     3636         16500         ? 0
 20         99   4223     3672         16500         ? 0
 30         89   2903     3708         16500         ? 0
 40         79   1055     3744         16500         ? 0
 50         68   3959     3780         16500         ? 0
 60         58   1055     3816         16500         ? 0
 70         47   2903     3852         16500         ? 200
 80         37   1883     3482.87      14500         ? 200
 90         28   1191     3086.71      12500         ? 200
 100        20   1251     2659.65      10500         ? 200
 110        13   2549     2196.95      8500          ? 200
 120        8    370      1692.63      6500          ? 100
 130        3    3778     1440.59      5500          ? 75
ON MOON AT 139.926 SECONDS - IMPACT VELOCITY 1253.25 MPH
SORRY THERE WERE NO SURVIVORS.  YOU BLEW IT!
IN FACT, YOU BLASTED A NEW LUNAR CRATER 347.15 FEET DEEP!

TRY AGAIN??
```

```
10 PRINT TAB(33);"LUNAR"
20 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
25 PRINT:PRINT:PRINT
30 PRINT "THIS IS A COMPUTER SIMULATION OF AN APOLLO LUNAR"
40 PRINT "LANDING CAPSULE.": PRINT: PRINT
50 PRINT "THE ON-BOARD COMPUTER HAS FAILED (IT WAS MADE BY"
60 PRINT "XEROX) SO YOU HAVE TO LAND THE CAPSULE MANUALLY."
70 PRINT: PRINT "SET BURN RATE OF RETRO ROCKETS TO ANY VALUE BETWEEN"
80 PRINT "0 (FREE FALL) AND 200 (MAXIMUM BURN) POUNDS PER SECOND."
90 PRINT "SET NEW BURN RATE EVERY 10 SECONDS.": PRINT
100 PRINT "CAPSULE WEIGHT 32,500 LBS; FUEL WEIGHT 16,500 LBS."
110 PRINT: PRINT: PRINT: PRINT "GOOD LUCK"
120 L=0
130 PRINT: PRINT "SEC","MI + FT","MPH","LB FUEL","BURN RATE":PRINT
140 A=120:V=1:M=33000!:N=16500:G=1E-03:Z=1.8
150 PRINT L,INT(A);INT(5280*(A-INT(A))),3600*V,M-N,:INPUT K:T=10
160 IF M-N<1E-03 THEN 240
170 IF T<1E-03 THEN 150
180 S=T: IF K>=N+S*K THEN 200
190 S=(M-N)/K
200 GOSUB 420: IF I<=0 THEN 340
210 IF V<=0 THEN 230
220 IF J<0 THEN 370
230 GOSUB 330: GOTO 160
240 PRINT "FUEL OUT AT";L;"SECONDS":S=(-V+SQR(V*V+2*A*G))/G
250 V=V+G*S: L=L+S
260 W=3600*V: PRINT "ON MOON AT";L;"SECONDS - IMPACT VELOCITY";W;"MPH"
270 IF W<=1.2 THEN PRINT "PERFECT LANDING!": GOTO440
280 IF W<=10 THEN PRINT "GOOD LANDING (COULD BE BETTER)":GOTO440
282 IF W>60THEN300
284 PRINT "CRAFT DAMAGE... YOU'RE STRANDED HERE UNTIL A RESCUE"
286 PRINT "PARTY ARRIVES.  HOPE YOU HAVE ENOUGH OXYGEN!"
288 GOTO 440
300 PRINT "SORRY THERE WERE NO SURVIVORS.  YOU BLEW IT!"
310 PRINT "IN FACT, YOU BLASTED A NEW LUNAR CRATER";W*.277;"FEET DEEP!"
320 GOTO 440
330 L=L+S: T=T-S: M=M-S*K: A=I: V=J: RETURN
340 IF S<5E-03 THEN 260
350 D=V+SQR(V*V+2*A*(G-Z*K/M)):S=2*A/D
360 GOSUB420: GOSUB 330: GOTO 340
370 W=(1-M*G/(Z*K))/2: S=M*V/(Z*K*(W+SQR(W*W+V/Z)))+.05:GOSUB 420
380 IF I<=0 THEN 340
390 GOSUB 330: IF J>0 THEN 160
400 IF V>0 THEN 370
410 GOTO 160
420 Q=S*K/M: J=V+G*S+Z*(-Q-Q*Q/2-Q^3/3-Q^4/4-Q^5/5)
430 I=A-G*S*S/2-V*S+Z*S*(Q/2+Q^2/6+Q^3/12+Q^4/20+Q^5/30):RETURN
440 PRINT:PRINT:PRINT:PRINT "TRY AGAIN??": GOTO 70
```

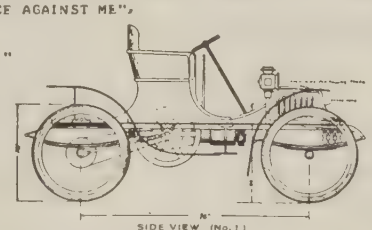
Drag

Language: BASIC (HP 2000C)

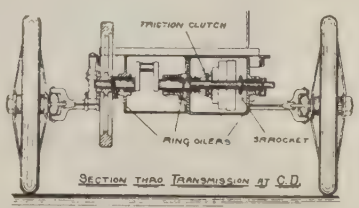
Description: DRAG allows the user to design his own dragster and then race it against a dragster designed by another player or the computer. You must specify the horsepower, rear end ratio, tire width, and tire diameter. There are no limits to these parameters.

Aha! you say. "I'll just design a two million horsepower dragster!" But it doesn't work that way, because your mass is related to your engine size, and so you usually end up with a top speed of something like 33 MPH. The computer is extremely hard to beat, but it's rumored that it can be done. Note: on some systems the amount of time between printouts can be aggravatingly long.

```
 LIST
DRAG

1   REM  ****  HP BASIC PROGRAM LIBRARY  ******************************
2   REM
3   REM          DRAG:   SIMULATES A DRAG RACE
4   REM
5   REM          36601 REV  B    10/73
6   REM
7   REM  ****  CONTRIBUTED PROGRAM  **********************************
10  DIM I$[3],P[2],E[2],L[2],S[2],X[2],M[2],C[2],B[2],Y[2]
20  DIM Q[2]
30  PRINT "WELCOME TO THE DRAG STRIP."
40  PRINT "WOULD YOU LIKE INSTRUCTIONS",
50  INPUT I$
60  IF I$="NO" THEN 110
70  PRINT "YOU MAY RACE AGAINST ONE OF YOUR FRIENDS OR YOU MAY RACE"
80  PRINT "AGAINST MY DRAGSTER.  YOU WILL BE ASKED TO DESIGN YOUR"
90  PRINT "OWN MACHINE, SPECIFYING HORSEPOWER, REAR END RATIO (X:1),"
100 PRINT "TIRE WIDTH IN INCHES AND TIRE DIAMETER IN FEET."
110 PRINT "DO YOU WANT TO RACE AGAINST ME",
120 INPUT I$
130 IF I$="NO" THEN 200
140 PRINT "I WILL HAVE CAR#1."
150 P[1]=600
160 E[1]=5.9
170 W[1]=22
180 D[1]=3.9
190 GOTO 290
200 PRINT "DESIGN CAR#1:"
210 PRINT "HORSEPOWER=",
220 INPUT P[1]
230 PRINT "REAR END RATIO=",
240 INPUT E[1]
250 PRINT "TIRE WIDTH=",
260 INPUT W[1]
270 PRINT "TIRE DIAMETER=",
280 INPUT D[1]
290 PRINT "DESIGN CAR#2:"
300 PRINT "HORSEPOWER=",
310 INPUT P[2]
320 PRINT "REAR END RATIO=",
330 INPUT E[2]
340 PRINT "TIRE WIDTH=",
350 INPUT W[2]
360 PRINT "TIRE DIAMETER=",
370 INPUT D[2]
380 PRINT
390 PRINT "GO!"
400 K1=500
410 K2=1.6
420 K3=2
430 K4=.0006
440 K5=.00006
450 K6=.2
460 K7=4
470 K8=.00015
480 Q[1]=Q[2]=0
490 S[1]=S[2]=0
500 X[1]=X[2]=0
510 REM:M IS MASS
520 FOR J=1 TO 2
530 M[J]=(K1+K2*P[J]+K3*W[J]*D[J]+K7*D[J]^2)/32.2
540 REM:C IS DRAG FROM WIND
550 C[J]=K4*M[J]^(2/3)+K8*W[J]*D[J]
560 REM:B IS THE MAX ACCELERATION WITHOUT BURNING
570 B[J]=15+28*W[J]*D[J]/ (L[J]+6)*(D[J]+1))
580 REM:Y IS THE SCALE FACTOR FOR RPM VS POWER
590 Y[J]=3.7-.0033*P[J]
600 NEXT J
610 PRINT
620 PRINT
630 PRINT "ELAPSED";TAB(15);"CAR#1";TAB(39);"CAR#2"
640 PRINT "TIME    SPEED       DISTANCE      SPEED       DISTANCE"
650 PRINT "(SEC)  (MPH)         (FT)        (MPH)         (FT)"
660 PRINT
670 FOR T=0 TO 100 STEP 1
680 FOR T1=1 TO 100 STEP 1
690 FOR J=1 TO 2
700 REM:R IS RPM
710 R=60*S[J]*E[J]/(3.14159*D[J])
720 REM:L0 IS ENGINE TORQUE
730 L0=(P[J]/42.5)*(50+.0078*(R/Y[J])-4.E-10*(R/Y[J])^3)
740 REM:L1 IS TORQUE FROM FRICTION
750 L1=P[J]*(K5*R+K6)
760 REM:L2 IS REAR AXEL TORQUE
770 L2=E[J]*(L0-L1)
780 REM F IS FORCE ON ROAD FROM TIRES
790 F=2*L2/D[J]
800 REM:TEST FOR BURN
810 IF F>M[J]*B[J] THEN 880
820 REM:A=ACCELERATION
830 IF Q[J] <> 0 THEN 860
840 PRINT "CAR#";J;"STOPS BURNING RUBBER"
850 Q[J]=1
860 A=(F-C[J]*S[J]^2)/M[J]
870 GOTO 900
880 A=B[J]-C[J]*S[J]^2/M[J]
890 REM:S=SPEED IN FT/SEC
900 S[J]=S[J]+A*.01
910 REM:X=DISTANCE IN FT
920 X[J]=X[J]+S[J]*.01
930 NEXT J
940 REM:TEST FOR FANISH
950 IF X[1]<5280/4 AND X[2]<5280/4 THEN 1160
960 IF X[1]>X[2] THEN 1080
970 T3=(X[2]-5280/4)/S[2]
980 T=T+T1/100-T3
990 X[2]=5280/4
1000  X[1]=X[1]-S[1]*T3
1010  PRINT T;S[1]*3600/5280;X[1];S[2]*3600/5280;X[2]
1020  PRINT TAB(40);"WINNER"
1030  PRINT
1040  PRINT "DO YOU WANT TO TRY AGAIN",
1050  INPUT I$
1060  IF I$="YES" THEN 110
1070  STOP
1080  T3=(X[1]-5280/4)/S[1]
1090  T=T+T1/100-T3
1100  X[1]=5280/4
1110  X[2]=X[2]-S[2]*T3
1120  PRINT T;S[1]*3600/5280;X[1];S[2]*3600/5280;X[2]
1130  PRINT TAB(10);"WINNER"
1140  PRINT
1150  GOTO 1040
1160  NEXT T1
1170  PRINT T+1;S[1]*3600/5280;X[1];S[2]*3600/5280;X[2]
1180  NEXT T
1190  STOP
1200  END

 RUN
DRAG

WELCOME TO THE DRAG STRIP.
WOULD YOU LIKE INSTRUCTIONS   ?NO
DO YOU WANT TO RACE AGAINST ME?YES
I WILL HAVE CAR#1.
DESIGN CAR#2:
HORSEPOWER=      ?370
REAR END RATIO=?5
TIRE WIDTH=      ?24
TIRE DIAMETER= ?2.7

GO!

ELAPSED          CAR#1                    CAR#2
TIME     SPEED        DISTANCE       SPEED        DISTANCE
(SEC)    (MPH)          (FT)         (MPH)          (FT)

  1      22.0707       16.3821       21.2787       15.7948
  2      43.5753       64.7826       42.0023       62.4532
  3      64.0015      143.982        61.6695      138.78
  4      82.9393      252.079        79.8806      242.914
  5     100.107       386.668        96.3631      372.503
  6     115.356       545.025       110.976       524.891
CAR# 1     STOPS BURNING RUBBER
  7     128.639       724.299       123.696       697.306
CAR# 2     STOPS BURNING RUBBER
  8     139.186       921.156       134.424       886.961
  9     146.922      1131.33        142.731      1090.55
 9.86047    151.81        1320        148.254       1274.32
          WINNER

DO YOU WANT TO TRY AGAIN      ?NO
```

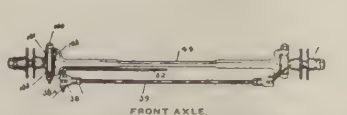


SIDE VIEW (No. 1)
Showing arrangement of Hanging Engine, etc.

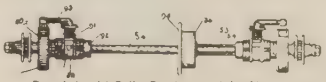


SECTION THRO TRANSMISSION AT C.D.

CRANK SHAFT SOLID from one end to the other. All key seats carefully milled.
2 RING OILER BEARINGS and a long bronze bearing in the center. Improved high speed friction clutch.



FRONT AXLE.
With Genuine Artillery Hubs and Timken Roller Bearings.



Rear Axle with Roller Bearings contained in case, ready for Springs and Radius

Side View No. 1 Engine and Transmission

Landing Simulator

Jake Jacobs

I have always wanted to learn how to fly an airplane but never took the time. After I had my Apple computer for a few weeks and had impressed myself with its capabilities, I undertook the task of developing a simple landing simulator in Applesoft Basic. To keep it simple and also to minimize the flicker on the screen I eliminated banking and turning. The controls are the "stick" which moves only forwards and backwards (not left or right) and the 'throttle' which controls power from 0 to 100 percent. Both controls are simulated by the game paddles. Paddle 0 is the stick and paddle 1 is the throttle.

The screen displays the pilot's view of the runway using hi-res graphics. The lower four text lines simulate the pilot's instruments. These are: ALTitude in feet, rate of CLIMB or descent, in feet/minute, VELocity in knots, DME (distance measuring equipment) which indicates the distance in nautical miles from the front end of the runway, and percent POWER applied by the throttle. There is also a glide-slope indicator which shows the pilot whether he or she is on the 3.5 degree glide slope when approaching the runway.

There are two principle parameters to be computed: distance from the runway (X) and altitude (ALT). The program is a continuous loop. During every pass through this loop the program calculates the new values of X and ALT from the old values. For example X is calculated from the old X by the statement X = X + V, where V is the current velocity. Similarly, the new altitude is calculated by adding the rate of climb to the old altitude. These values are derived from the two input variables, the stick and the throttle. Once the values of X and ALT are computed, the view of the runway as seen by the pilot is then calculated and displayed in hi-res graphics. But first the old image of the runway is erased by redrawing it with HCOLOR set equal to 0.

I undertook the task of developing a simple landing simulator in Applesoft Basic.

The runway appears as a trapezoid in most cases. Since banking and turning are not simulated, the view of the runway is always symmetrical about an imaginary vertical line down the center of the screen (see Photo 1). Therefore only four values have to be calculated: the vertical positions of the horizontal lines which simulate the rear and front of the runway, and the widths of the rear and front of the runway.

Interestingly, no trigonometric functions have to be used. Only the simple geometry of similar triangles is applied. Figure 1 depicts a side view of the aircraft and the runway. The pilot's eye is assumed to be one foot away from the windshield

Jake Jacobs, 1903 Fordham Way, Mountain View, CA 94040.

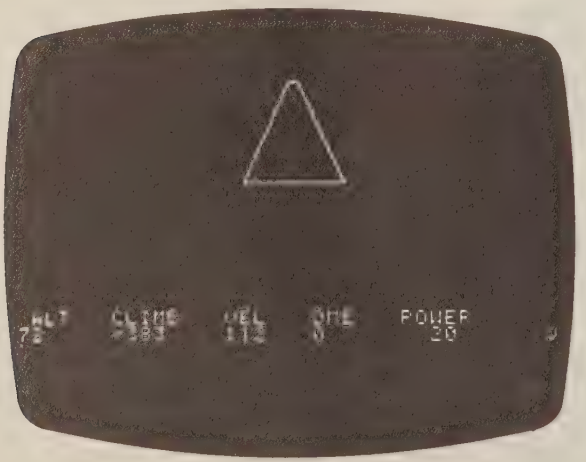


Photo 1. View of the runway as the aircraft is about to land. Note the negative rate of CLIMB.

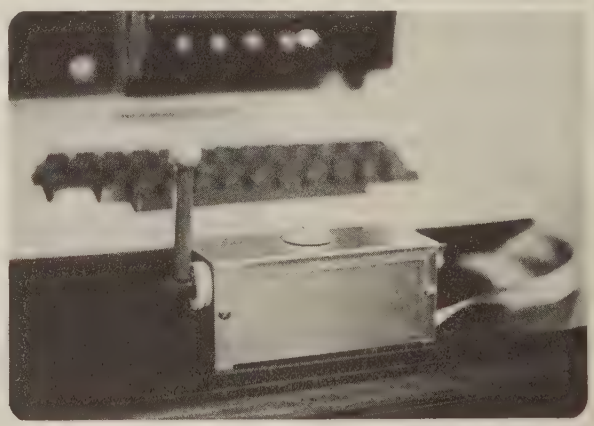


Photo 2. Photograph of author's "stick" and "throttle" box which replaces the game paddles. However the game paddles work just fine so you need not build your own box.

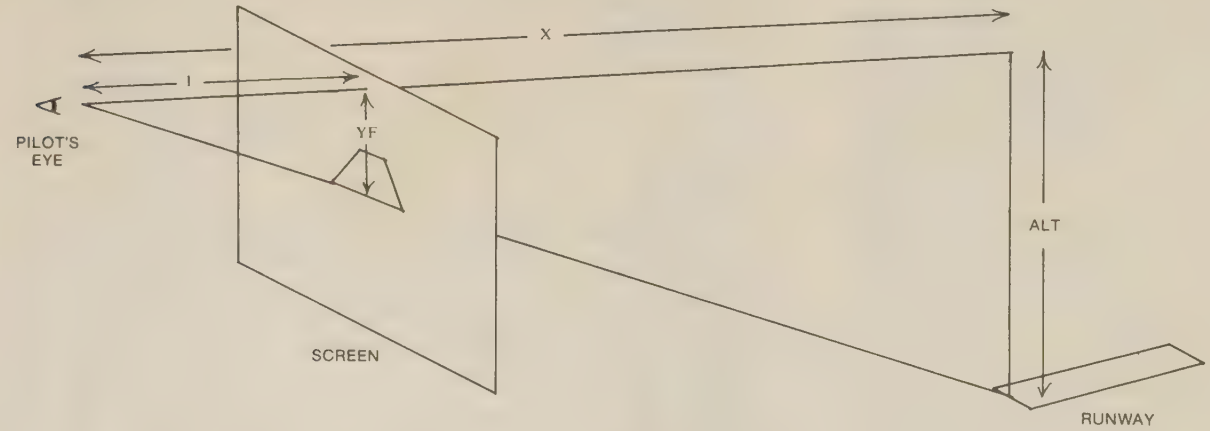


Figure 1. Side view of aircraft illustrating how the screen position of the front of the runway, YF, is calculated. Only simple similar triangles are used.

(i.e., the TV monitor, but you don't have to sit that close to your screen; the one foot is for calculation purposes only). Therefore the Y offset from the top of the screen (YF) of the front of the runway view is calculated from similar triangles as follows:

$$\frac{YF}{1} = \frac{ALT}{X}$$

YF must be scaled by multiplying it by 159, the maximum Y value that can be HPLOTed. If you have followed this so far, you will realize that for some values of X and ALT, YF could have a value that would plot off the bottom of the screen. In fact much of the program loop is taken up calculating what parts of the view of the runway are off the screen and correcting for this so that the HPLOT statements do not "blow up."

In a similar fashion, YR, the Y offset of the rear of the runway, is calculated. The same principle of similar triangles is employed to calculate the widths of the front and rear of the runway, and will not be described here.

A word about programming. Note that all constants used in the program are actually variables that are initialized at the beginning of the program. This was done for three reasons.

1. During run-time, variables are accessed faster than the time it takes for numeric constant strings to be evaluated.

2. When writing the program it was much easier to change a single assignment statement at the beginning of the program than to find all occurences of a particular constant throughout the program.

3. Using variables instead of numeric constants makes the program much more readable and self documenting. Note that some constants have the same value as others, but are given different names. (Remarks were included in the original program but many were deleted for this article to allow the program to run on a 16K machine.)

Here we introduce exponential smoothing, which is a fancy term for something quite simple.

If, for example, one wished to change the length of the runway from 4000 feet to 6000 feet, one would merely change the assignment statement 280 to RW = 6000. All occurences of the "constant" RW (and there are many) will be changed correctly.

Flying the Airplane

You start at a random distance from the runway and at a random altitude. The runway starts out as a small dot or line on the screen. As you approach the runway it will become larger and take on the appearance of a trapezoid. The program is calculating what the runway should really look like from your given altitude and distance from the runway. The throttle controls primarily your rate of climb and descent. The stick controls primarily your airspeed but also your rate of climb and descent. This is just the opposite of what most people think, but that is how an airplane really works. The best way to "fly" the plane is to place paddle 1 (throttle) on the table and put the stick (paddle 0) in your left hand and control it with your right hand. Then you can reach up to the table to adjust the throttle with your right hand. The author has built a "stick" and "throttle" in a small metal box as shown in Photo 2, but the game paddles work just fine.

The ALTitude and CLIMB indicators work together. Your altitude is shown in feet (from the ground) and the CLIMB indicator shows your rate of climb in feet per minute if positive or descent if negative. The VELocity indicator shows airspeed in knots or nautical miles per hour. A nautical mile is 6000 feet. The DME indicator is the Distance Measuring Equipment and indicates your distance from the front end of the runway in nautical miles. After you pass over the front of the runway the DME will begin to increase in value, rather than decrease as you approach the runway, because you are moving away from the front of the runway. The POWER indicators merely parrots the throttle setting from 0 to 100 percent. The last indicator on the right is the glide-slope indicator. Imagine an invisible line with a 3.5 degree slope raising towards you from the front of the runway. As you land you should stay on this imaginary line. The glide-slope indicator tells you if you are on this line and if not, what you should do:

- – means you are on the glide-slope
- ^ means that you should climb (you are too low)
- V means that you should go down (you are too high).

When you are very close to the runway, you should ignore the glide-slope indicator since it applies to the front of the runway and you never actually land at the very front of the runway.

There are three markers on the runway, one every 1000 feet. These give you some idea how much runway you have left when landing. If you decide to change the length of the runway (see below), the markers will space themselves correctly ¼, ½ and ¾ the distance down the runway.

When cruising your throttle should be set at about 75 percent. When climbing you should be at 100 percent throttle. When landing, 20 percent is about right. You must land with a descent of less than 150 feet/minute, otherwise you will crash. If your velocity falls below 60 knots you will stall and your rate of descent will become -2000 feet/minute. But you can recover by pushing the stick foward to increase your lift. You can crash in other ways. If you hit the ground before you are over the runway you have crashed. If you touchdown on the runway but there is too little runway remaining you will crash off the end of the runway. And of course you can always fly over the runway and not touch down at all. All of these conditions are indicated by messages on the screen.

The Program

Lines 10 through 410 initialize variables that are used as constants. Some that might be of interest if you wish to modify the program are Line 310, VM — stall velocity, Line 380, AM — minimum altitude, below which you are considered to be on the ground, and Line 400, RC — crash descent, above which you have landed too hard. Lines 450 and 460 establish the initial values for X and ALT. You may choose to change the algorithms for these initial conditions.

Line 500 starts the main loop of the program, which runs down to line 840. This loop is for a trapezoidal runway view. There are a couple of branches to lines 850 or 1040 if the runway view is partially off of the screen. The main loop calculates and plots the runway view. Subroutine 1200 is the main subroutine in the program. It sets P0 and P1 to the paddle values and uses these to calculate a new velocity, V, from the old V. Here we introduce exponential smoothing, which is a fancy term for something quite simple. Smoothing creates lag or inertia into some variables. For example, the velocity should not change instantaneously (nor should the rate of change of altitude or the pitch angle of the aircraft). These variables are permitted to change only gradually. Let us look at a simplified version of line 1240 for calculating V.

$$V = .95\ V + .05\ DV$$

where DV is the change of velocity (based on the stick and throttle positions) and is used here for illustrative purposes only. The new velocity is the old velocity plus the change in velocity, DV. But we only let the DV term influence 5 percent of the new velocity and let the old velocity influence

```
0  REM  VARS. FO, HO, RO, XO: SECOND CHARACTER IS A LETTER "OH"; PO: SECOND
 CHARACTER IS DIGIT "ZERO"
10  TEXT : HOME
20  GOSUB 1700
30  REM CONSTANTS
40 CZ = 2
50 C1 = 0
60 C2 = .03
70 C3 = .7
80 C4 = .09
90 C5 = .39
100 C8 = 30
110 C9 = .09
120 DC = 100
130  REM EXPONENTIAL SMOOTHING
140 D = .95
150 DD = 1 - D
160 E = .8
170 EE = 1 - E
180 F = .8
190 FF = 1 - F
200 H1 = .3
210 P5 = .055
220 P7 = .065
230  REM HI-RES CONSTANTS
240 CE = 139.5
250 VE = 191
260 HO = 279
270  REM RUNWAY LENGTH
280 RW = 4000
290 M2 = RW / 2:M1 = RW / 4:M3 = M1 + M2
300  REM STALL VEL.
310 VM = 60
320 VC = 1
330 ST = 20
340 Q1 = 255
350 Q3 = 3
360 Q6 = 600
370  REM STALL DESCENT
380 AM = 5
390  REM CRASH DESCENT
400 RC = 150
410 C = 7000
420  REM INITIAL CONDX.
430 V = 100
440 HR = H1 * (Q1 -  PDL (0))
450 X = 20000 + 20000 *  RND (1)
460 ALT = 1000 + 1000 *  RND (1)
470 X1 = 0:X2 = X1:X3 = X1:X4 = X1:Y1 = X1:Y2 = Y1:Y3 = Y1:Y4 = Y1:X5 = X1:Y5
 = Y1
480  HGR
490  HOME
500  REM START MAIN LOOP
510  IF ALT < AM THEN 1400
520 YR = ALT / X * VE + HR
530  IF YR > VE OR YR < 0 THEN YR = VE
540 YF = ALT / (X - RW) * VE + HR
550 F1 = 0:F2 = F1:F3 = F1
560 R1 = ALT / (X - M1) * VE + HR: IF R1 > 0 AND R1 < VE AND X > M1 THEN
 F1 = 1
570 R2 = ALT / (X - M2) * VE + HR: IF R2 > 0 AND R2 < VE AND X > M2 THEN
 F2 = 1
580 R3 = ALT / (X - M3) * VE + HR: IF R3 > 0 AND R3 < VE AND X > M3 THEN
 F3 = 1
590  IF X - RW < ALT THEN YF = VE
600 FO = (C / (X + ALT / Q3 - RW))
610 RO = C / (X + ALT / Q3)
620  IF X - RW < ALT THEN FO = C / (ALT + ALT / Q3)
630  IF RO < 0 THEN RO = CE
640 FL = CE - FO
650 FR = CE + FO
660 RL = CE - RO
670 RR = CE + RO
680  IF RO >  = CE GOTO 1570
690  IF FO > CE THEN 850
700  IF YF > VE THEN 1040
710  HCOLOR= 0
720  IF E1 THEN  HPLOT CE,N1
730  IF E2 THEN  HPLOT CE,N2
740  IF E3 THEN  HPLOT CE,N3
750  HPLOT X1,Y1 TO X2,Y2 TO X3,Y3 TO X4,Y4 TO X1,Y1: IF Y5 < VE THEN  HPLOT
 0,Y5 TO HO,Y5
760  HCOLOR= COL
770  IF F1 THEN  HPLOT CE,R1
780  IF F2 THEN  HPLOT CE,R2
790  IF F3 THEN  HPLOT CE,R3
800  HPLOT RL,YR TO RR,YR TO FR,YF TO FL,YF TO RL,YR
810 X1 = FL:Y1 = YF:X2 = RL:Y2 = YR:X3 = RR:Y3 = YR:X4 = FR:Y4 = YF:Y5 =
 VE
820 N1 = R1:N2 = R2:N3 = R3:E1 = F1:E2 = F2:E3 = F3
```

```
830  GOSUB 1200
840  GOTO 500
850  REM
860 YB = YF - (YF - YR) * (FO - CE) / (FO - RO)
870  IF YB < 0 THEN YB = 0
880  IF YB > VE THEN 1040
890  HCOLOR= 0
900  IF E1 THEN  HPLOT CE,N1
910  IF E2 THEN  HPLOT CE,N2
920  IF E3 THEN  HPLOT CE,N3
930  HPLOT X1,Y1 TO X2,Y2 TO X3,Y3 TO X4,Y4 TO X1,Y1: IF Y5 < VE THEN  HPLOT
 0,Y5 TO HO,Y5
940  HCOLOR= COL
950  IF F1 THEN  HPLOT CE,R1
960  IF F2 THEN  HPLOT CE,R2
970  IF F3 THEN  HPLOT CE,R3
980  HPLOT 0,YB TO RL,YR TO RR,YR TO HO,YB
990  IF YF < VE THEN  HPLOT 0,YF TO HO,YF
1000 X1 = 0:Y1 = YB:X2 = RL:Y2 = YR:X3 = RR:Y3 = YR:X4 = HO:Y4 = YB:Y5 =
 YF
1010 N1 = R1:N2 = R2:N3 = R3:E1 = F1:E2 = F2:E3 = F3
1020  GOSUB 1200
1030  GOTO 500
1040  REM
1050 XO = (VE - YR) * (FO - RO) / (YF - YR) + RO
1060  HCOLOR= 0
1070  IF E1 THEN  HPLOT CE,N1
1080  IF E2 THEN  HPLOT CE,N2
1090  IF E3 THEN  HPLOT CE,N3
1100  HPLOT X1,Y1 TO X2,Y2 TO X3,Y3 TO X4,Y4 TO X1,Y1: IF Y5 < VE THEN  HPLOT
 0,Y5 TO HO,Y5
1110  HCOLOR= COL
1120  IF F1 THEN  HPLOT CE,R1
1130  IF F2 THEN  HPLOT CE,R2
1140  IF F3 THEN  HPLOT CE,R3
1150  HPLOT CE - XO,VE TO RL,YR TO RR,YR TO CE + XO,VE
1160 X1 = CE - XO:Y1 = VE:X2 = RL:Y2 = YR:X3 = RR:Y3 = YR:X4 = CE + XO:Y4
 = VE
1170 N1 = R1:N2 = R2:N3 = R3:E1 = F1:E2 = F2:E3 = F3
1180  GOSUB 1200
1190  GOTO 500
1200  REM MAIN SUBROUTINE
1210 P1 =  PDL (1)
1220 PW =  INT (P1 * C5)
1230 PO =  PDL (0)
1240 V = D * V + DD * (PO * C3 + C8 + C4 * P1)
1250 CL = DC * DALT
1260 ALT = ALT + DALT
1270 VV = V * VC
1280  VTAB 22: HTAB 1
1290  PRINT  TAB( 2);"ALT"; TAB( 8);"CLIMB"; TAB( 16);"VEL"; TAB( 22);"DME";
 TAB( 28);"POWER"
1300  VTAB 23: HTAB 1
1310 AR$ = "-"
1320  IF ALT > P7 *  ABS (X - RW) THEN AR$ = "V"
1330  IF ALT < P5 *  ABS (X - RW) THEN AR$ = "^"
1340  PRINT  TAB( 1); INT (ALT); TAB( 8); INT (CL); TAB( 16); INT (VV); TAB(
 22); INT ( ABS ((X + ALT / Q3 - RW) / Q6)) / 10; TAB( 30);PW;"  "; TAB(
 38);AR$
1350 X = X - V
1360 DALT = E * DALT + EE * (C2 * P1 - V * C1 - C9 * PO + CZ)
1370  IF V < VM THEN DALT =  - ST
1380 HR = F * HR + FF * H1 * (Q1 - PO)
1390  RETURN
1400  REM LANDED!
1410  IF X > RW GOTO 1510
1420 X = X - 10 * V
1430  IF X < 0 GOTO 1630
1440  IF CL <  - RC GOTO 1780
1450  PRINT "YOU LANDED AT "; INT (VV);" MPH AND STOPPED "
1460  PRINT  INT (X);" FEET FROM THE END OF THE RUNWAY."
1470  PRINT "YOU WERE DESCENDING AT "; -  INT (CL);" FEET/MINUTE."
1480  GET Z$
1490  TEXT : HOME
1500  GOTO 20
1510  REM
1520  TEXT : HOME
1530  GOSUB 1850
1540  PRINT "YOU CRASHED "; INT (X - RW);" FEET SHORT"
1550  PRINT "OF THE RUNWAY AT "; INT (VV);" MPH. TRY AGAIN"
1560  GOTO 20
1570  REM
1580  TEXT : HOME
1590  PRINT "YOU OVERFLEW THE RUNWAY AT "; INT (VV)
1600  PRINT "MPH AT AN ALTITUDE OF "; INT (ALT)
1610  PRINT "FEET.  TRY AGAIN!!!"
1620  GOTO 20
1630  REM
1640  TEXT : HOME
1650  GOSUB 1850
```

the new velocity 95 percent. This puts lag into the rate at which the velocity can change which tends to simulate the real world. You can change the smoothing constants in lines 140, 160, and 180.

The main subroutine calculates the instrument values and displays them in line 1340. The pitch angle of the aircraft is simulated by variable HR in line 1380 which offsets the view of the runway on the screen. This offset calculation is highly simplified and is a function of the stick only. In a real aircraft the pitch angle is a function of airspeed, throttle, and other variables. You may wish to try a more realistic expression for HR here.

Finally, there are a number of terminating messages. You have "landed" when your altitude is less than AM (line 510). Line 1400 begins the checks of what kind of landing you made (normal or crash). Line 1410 checks if you are short of the runway. Line 1420 assumes it takes ten times your velocity to stop and then line 1430 checks if you flew off the end of the runway. Finally line 1440 checks if you landed too hard. If you get past these checks, lines 1450 through 1470 print the safe landing message. Line 680 checks to see if you have flown off the end of the runway.

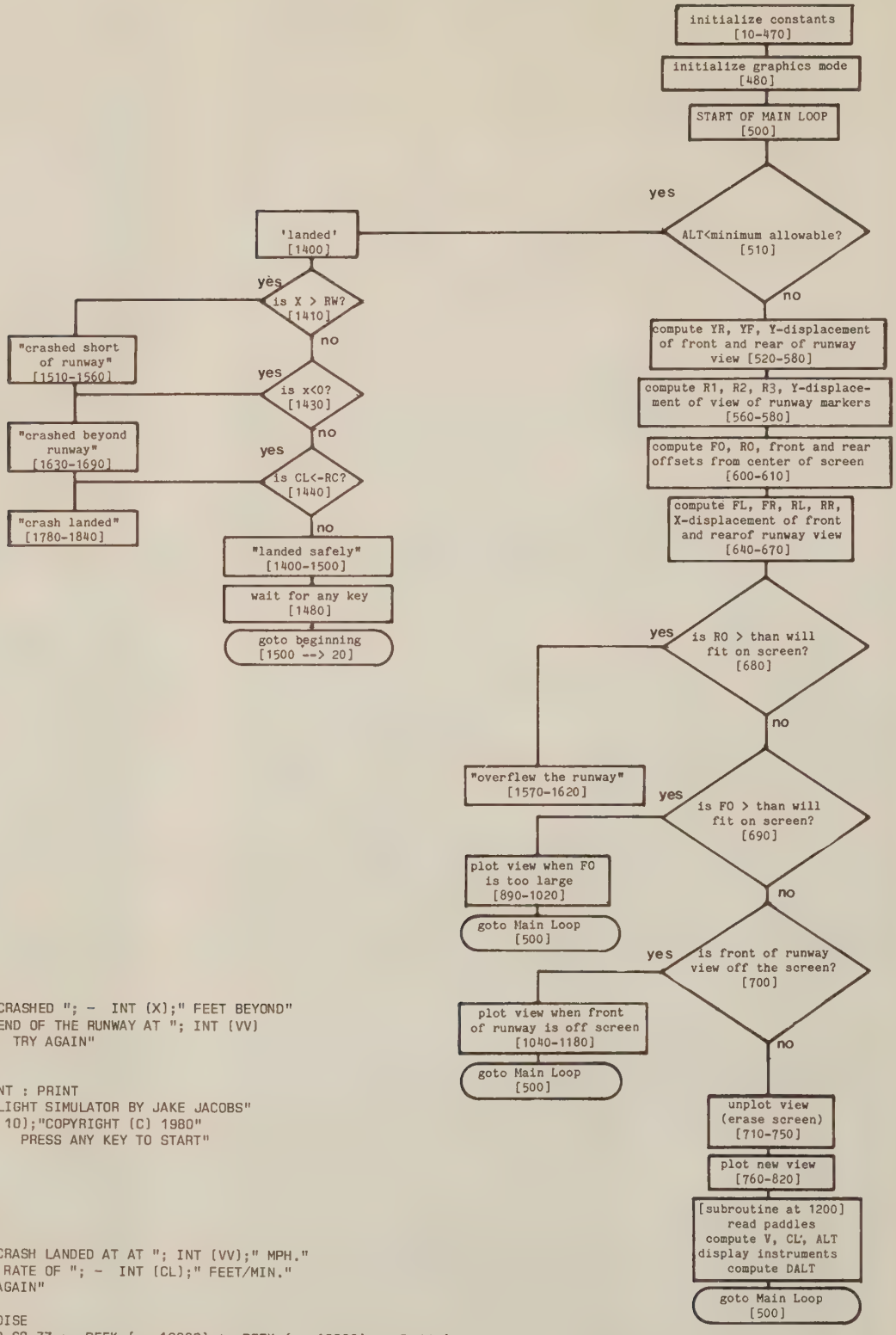
Program Modifications

The author has made many variations of this program such as adding instrument approach (where you cannot see the runway until you are very close to it), visual only approach (where the instruments do not work), a visual marker two miles from the runway, a marker which flashes an indicator when the plane is five or a half mile from the runway, and so on. One variation allows you to take off from one runway and land at another. You may want to experiment with such variables as

RW — runway length (4000 feet)
X — starting distance from runway
ALT — starting altitude
V — velocity (make a jet!)

Try changing various constants in expressions for V and DALT (rate of climb). (Do not delete any remarks in the program because some remarks are GOTO targets.)

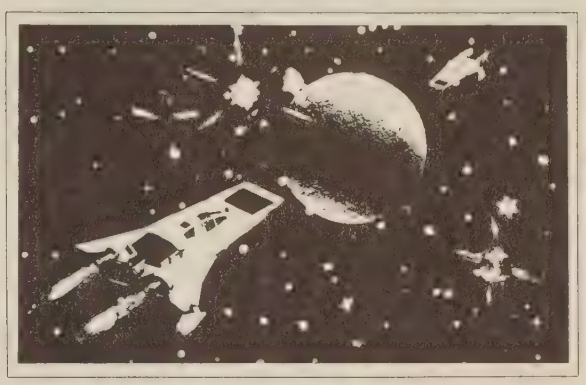
Sublogic Company (201 W. Springfield Ave., Champaign, IL 61820) has on the market the FS-1 Flight Simulator which is fantastic. It simulates turns, banks, and much more. I highly recommend it if you want to "fly." But since FS-1 is written in assembly language with no listings supplied, you cannot learn much about how it works (although their rather extensive operating manual has block diagrams of the simulator's structure). I started writing my Flying Simulator before I had heard of FS-1, but have to admit I spend more time "flying" FS-1 than my own. Happy landings. □




```
1660  PRINT "YOU CRASHED "; -  INT (X);" FEET BEYOND"
1670  PRINT "THE END OF THE RUNWAY AT "; INT (VV)
1680  PRINT "MPH.  TRY AGAIN"
1690  GOTO 20
1700  REM
1710  PRINT : PRINT : PRINT
1720  PRINT "   FLIGHT SIMULATOR BY JAKE JACOBS"
1730  PRINT  TAB( 10);"COPYRIGHT (C) 1980"
1740  PRINT "       PRESS ANY KEY TO START"
1750  GET Z$
1760 COL = 3
1770  RETURN
1780  REM
1790  TEXT : HOME
1800  GOSUB 1850
1810  PRINT "YOU CRASH LANDED AT AT "; INT (VV);" MPH."
1820  PRINT "AT A RATE OF "; -  INT (CL);" FEET/MIN."
1830  PRINT "TRY AGAIN"
1840  GOTO 20
1850  REM CRASH NOISE
1860  FOR I = 1 TO 60:ZZ =  PEEK ( - 16336) +  PEEK ( - 16336) -  PEEK (
 - 16336) -  PEEK ( - 16336): NEXT : RETURN
1870  END

]
```

Zeno's Spaceships

Michael J. Orlove



Two spacecraft are on a course for a head-on collision. They are both out of fuel but are speedily coasting toward each other on their own inertia.

They have greenhouses and farms on board and their inhabitants could go on living happily for generations if only the impending disaster could be prevented.

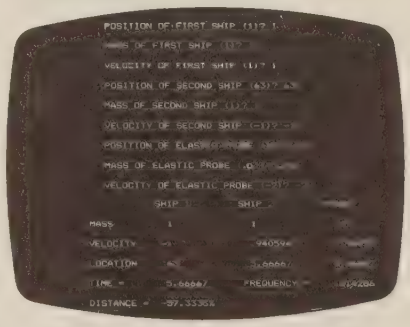
With this purpose in mind a ping-pong ball is fired from the first ship to the second. It bounces back and forth repeatedly between the ships. Every time the ball hits a ship the ship is slowed down a little and the ball is speeded up a little. What will happen?

The following program simulates this event.

One can imagine the ball bouncing between two parallel walls on the two ships, or alternatively, between two corner cubes. Thus the ball is not deflected sideways into space.

Zeno, an Ancient Greek mathematician, considered Achilles, a warrior, overtaking a tortoise in a race. His mathematical approach resembled a variable speed, slow motion camera which starts out at a reasonable speed but goes faster and faster having used an infinite length of film by the time Achilles overtakes the animal.

Zeno did the equivalent of playing this film on a constant speed projector in order to convince his poor audience that Achilles never catches up to the tortoise.


POSITION OF FIRST SHIP (1)? 1
VELOCITY OF FIRST SHIP (1)? 1
POSITION OF SECOND SHIP (43)? 43
MASS OF SECOND SHIP (1)? 1
VELOCITY OF SECOND SHIP (-1)? -1
MASS OF ELASTIC PROBE .0
VELOCITY OF ELASTIC PROBE -2
SHIP 1
SHIP 2
MASS 1 1
VELOCITY -1 .940594
LOCATION 1 43.0000
TIME = 5.0000 FREQUENCY = 1.04286
DISTANCE = 37.33366

Variable Initialization.

The program here displays one frame per ship/ball collision. Thus, when the ships are close and the ball is buzzing between them, we are in very slow motion. To keep a sensible perspective in mind, the display is a time-space graph.

Time is vertical on your screen and space is horizontal.

If you assign a mass of zero to the ping-pong ball, then an infinite number of ball/ship collisions will occur before the ship/ship collision. But this special case is indeed Zeno's case and you will wait forever for the ship/ship collision as your computer runs one frame per ship/ball collision.

For very small mass balls the simulation will not take forever but may still take a long time! But not if you use the "special feature": on request the program provides a feature which speeds the simulations.

This speed-up is accomplished by replacing the ping-pong ball you requested with a bigger, slower one. This allows the computer to plot a practically identical looking graph with fewer ball/ship collisions. The approximation is rendered best possible when the computer assigns to the substitute ping-pong ball the same initial energy as the previous one. This is done by saying: if the new ball is 1/X as fast as the old it must be X^2 times as heavy. For example a ball 1/4 as fast must be 16 times heavier and a ball 1/3 as fast

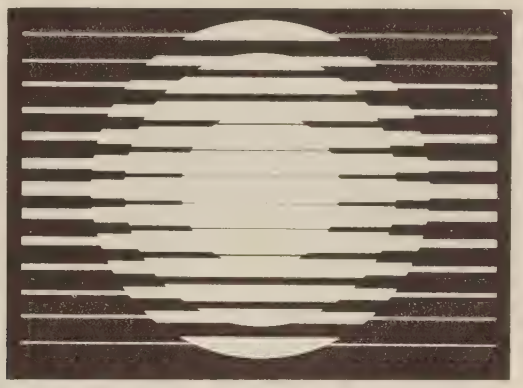
Michael J. Orlove, 2058 Powell Ave., Bronx, NY 10472.

must be 9 times heavier to have the same energy.

In conclusion, let me state that the ability to simulate continuous time as discrete events, as here, opens a can of philosophical worms. It suggests that real acceleration and deceleration may occur in steps as here.

If, however, position velocity and acceleration and all their derivatives and integrals are truly continuous, then all events past, future and distant could be known from a detailed record of one cubic meter of the universe spanning one second of history. □

```
ZENO'S SPACESHIPS
          BY MIKE ORLOVE
    CREATIVE COMPUTING VERSION"
3 PRINT"
   PARENTHESES GIVE SUGGESTED INPUTS"
10 DIM V$(63)
20 INPUT"DO YOU WANT SPEED FEATURE
   (NO)";A$
30 C4=0
40 INPUT"
AT WHAT TIME SHOULD VELOCITY STATISTICS
   FLASH ON SCREEN (1)";C
70 INPUT"POSITION OF FIRST SHIP (1)";A2
90 INPUT"MASS OF FIRST SHIP (1)";M1
110 INPUT"VELOCITY OF FIRST SHIP (1)";V1
130 INPUT"POSITION OF SECOND SHIP
    (63)";B2
150 INPUT"MASS OF SECOND SHIP (1)";M2
170 INPUT"VELOCITY OF SECOND SHIP
    (-1)";V2
190 A1=1
200 B1=1
210 Q1=1
220 INPUT"POSITION OF ELASTIC PROBE
    (15)";Q2
240 INPUT"MASS OF ELASTIC PROBE
    (.01)";M3
260 INPUT"VELOCITY OF ELASTIC PROBE
    (-2)";V3
280 IF A$="NO"THEN 320
290 E2=M3*V3[2
300 V3=(E2/(V1*.01))[.5*(-1*ABS(V1)/V1)
310 M3=M1*.01
320 I7=0
330 I7=I7+1
340 IF B1>C THEN 680
350 GOSUB 570
400 I1=A2-A1*V1
410 I3=Q2-Q1*V3
420 C1=(I1-I3)/(V3-V1)
430 C2=V1*C1+I1
440 V4=(M1*V1+M3*V3)/(M1+M3)
450 I4=C2-C1*V4
460 V3=2*V4-V3
470 V1=2*V4-V1
480 Q2=C2
490 Q1=C1
500 A1=B1
510 A2=B2
520 B1=C1
530 B2=C2
540 Z=M1
541 M1=M2
542 M2=Z
550 Z=V1
551 V1=V2
552 V2=Z
560 GOTO 330
570 IF INT(A1)=INT(B1) THEN 670
580 IF B2<1THEN 670
590 IF B2>63 THEN 670
600 IF INT(B1)-INT(A1)=1THEN 640
610 FORJ=A1TO B1-1
620 PRINT
630 NEXTJ
640 REM
650 FORI=1TOB2
651 PRINT" ";
652 NEXTI
653 PRINT". "
660 REM
670 RETURN
680  IF C4=1 THEN 350
690 PRINT ,"SHIP 1","SHIP 2",  "PROBE"
700 PRINT "MASS",M1,M2,M3
710 PRINT "VELOCITY",V1,V2,V3
720 PRINT"LOCATION",A2,B2,Q2
730 PRINT"TIME = ",B1,"FREQUENCY =
    ",1/(B1-A1)
740 PRINT "DISTANCE =",B2-A2
750 C4=1
755 INPUT "<<PRESS ENTER>>";Z$
760  GOTO 350
```



Planet of the Double Sun

Caxton C. Foster

In many a science fiction story the protagonists find themselves on a planet circling a double or triple sun. The strange multiple shadows, perhaps of different colors, and the spectacle of the suns orbiting around each other provide good and undeniably alien descriptive material for the author, who often gets paid by the word. For many years it was believed that in actuality no stable planetary orbit could exist in a multistar system. Since science fiction is fiction, the authors went ahead calmly with their stories emphasizing the fiction at the presumed expense of the science. Recently, it has been discovered that fiction was not as fictitious as was believed and that indeed there can be stable planets orbiting multiple suns.

New computer studies have shown (1) that if our sun were replaced by two stars 0.4 astronomical units apart (an astronomical unit is equal to the mean distance of the earth from the sun, approximately 93 million miles) each with mass and luminosity of half that of our sun, all the planets except Mercury would survive with orbits almost unchanged and the earth would receive approximately the same amount of radiation as it does now and, hence, would remain at the same temperature it now has.

A second example replaced Jupiter with a star equal in mass and luminosity to the sun. Mars would be lost this time, but the other planets including the earth would remain. Again the earth would not change its temperature substantially.

Figuring that a home computer occasionally ought to be permitted to do a computation and not always be forced to play host to assorted demons and dragons I set out

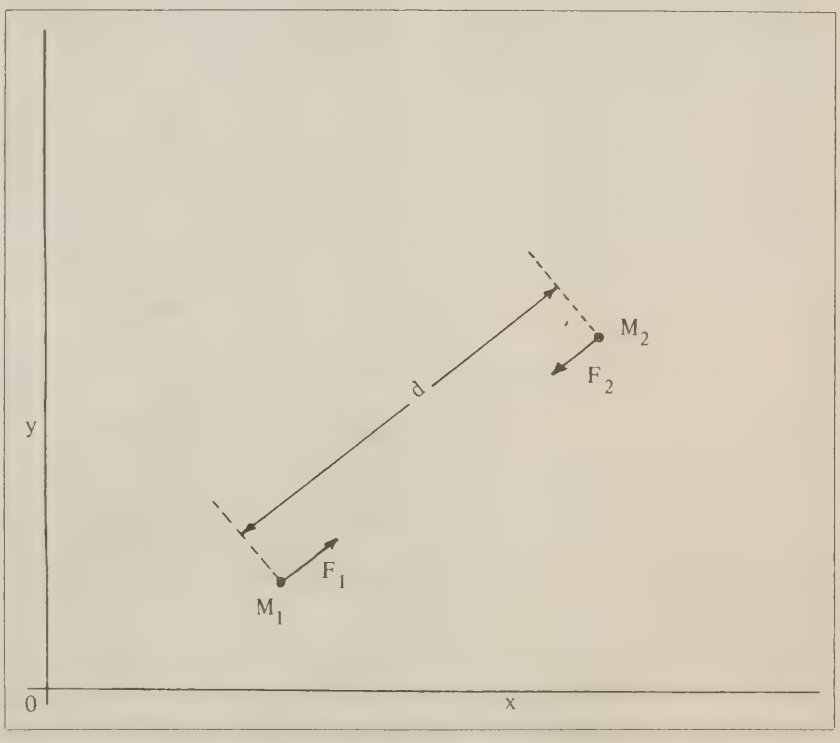


Figure 1. Two bodies in the x-y plane.

Caxton C. Foster, Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.

to build a model of a solar system with one planet and two suns.

Orbital Mechanics

It has been over thirty years since I studied mechanics and orbital dynamics and all I could remember was that every object in the universe attracts every other object directly proportional to the product of their masses and inversely as the square of the distance between them. Thus, if M_1 and M_2 are two masses at distance d from each other the force drawing them together is proportional to:

$$F = k \frac{m_1 \cdot m_2}{d^2}$$

Since we are interested only in general models we will set the constant of proportionality K equal to one. Dredging yet deeper in the river of time I recalled that force equals mass times acceleration, or:

$$F = ma, \text{ or}$$

$$a = \frac{F}{m}$$

So the acceleration that mass 1 will undergo will be:

$$a_1 = \frac{F}{m_1} = \frac{m_2}{d^2}$$

proportional to the mass of the other object divided by the distance squared.

Figure 1 shows two bodies in space. We assume that they are confined to the plane of the paper. The distance d separating them can be calculated using the theorem of Pythagoras:

$$d = \sqrt{(X_1-X_2)^2 + (Y_1-Y_2)^2}$$

To simplify our calculations we need to resolve the force F into its X and Y components. See Figure 2. The large triangle is formed by points M_1 and M_2 and lines ΔX parallel to the X-axis and ΔY parallel to the Y-axis. The length of these sides is given by:

$$\Delta X = (X_2-X_1), \text{ and}$$

$$\Delta Y = (Y_2-Y_1)$$

We want to separate F into its components F_x and F_y. The two triangles are similar so we can say that

$$F_x = F \cdot \frac{\Delta X}{d}$$

$$F_y = F \cdot \frac{\Delta Y}{d}$$

Now that the force has been resolved we can also resolve the acceleration into two components parallel to the X and Y axes as:

$$A_x = F_x/M$$

$$A_y = F_y/M$$

Knowing the acceleration in the X direction we can make the approximation that over a short time interval dt the force is constant (because the masses don't move very much if dt is small).

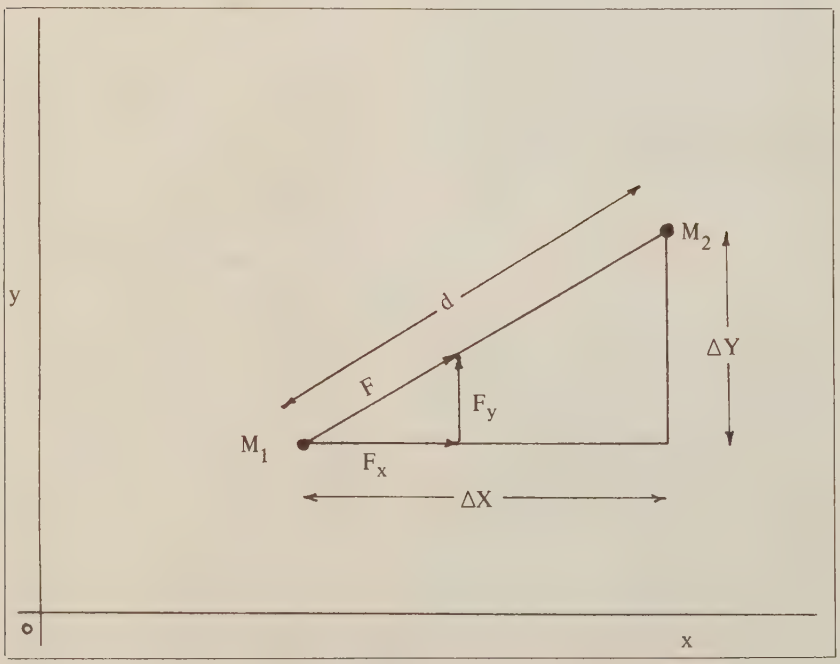


Figure 2. Resolution of force F into its X and Y components.

Thus: $V_x = V_x + A_x \cdot dt$

$$V_y = V_y + A_y \cdot dt$$

Now we further assume that for the short interval dt the velocity (V_x and V_y) is constant so

$$X = X + V_x \cdot dt$$

$$Y = Y + V_y \cdot dt$$

If we make dt very small these approximations will be almost exactly true, but we will have to do a large number of calculations to get any noticeable movement of the masses. A reasonable compromise between accuracy and speed seems to be found in the range of dt between 0.1 and 1.

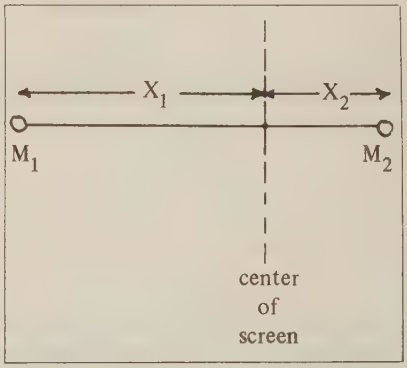


Figure 3. Bringing the center of mass to the center of the screen.

When there are two masses to work with and they are of comparable size they will orbit about the center of mass of the system. If we don't want the center of mass and, hence the two suns, to wander off the screen we must make certain that the initial conditions we give the system (the initial values for X_1, Y_1, X_2, Y_2, V_{x1}, V_{y1}, V_{x2}, and V_{y2}) do not contain a net motion in any direction.

I chose to begin my program with each body lying on the X-axis moving strictly vertically. Therefore, $Y_1 = Y_2 = V_{x1} = V_{x2} = 0$. To vary the initial conditions I read in X_1 and V_{y1}, the position and velocity of body 1. To bring the center of mass to the center of the screen we can look at Figure 3 and calculate that

$$|M_1 \cdot X_1| = |M_2 \cdot X_2|$$

$$X_2 = -X_1 \frac{M_1}{M_2}$$

Similarly to bring the net velocity of the center of mass to zero we have

$$V_{y2} = -V_{y1} \cdot \frac{M_1}{M_2}$$

The Three Body Problem

Two bodies rotating about each other *define* a plane in space. For simplicity and also to reflect the basically planar nature of our solar system we are going to assume that our three bodies move in a common plane.

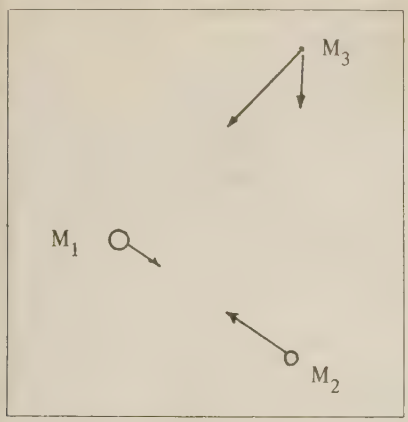


Figure 4. The two suns attract the planet and each other.

Now that we have seen how to handle the two suns we can introduce the planet. We assumed in the program that sun 1 had unit mass and express the mass of the second sun as a multiple of this unit. We

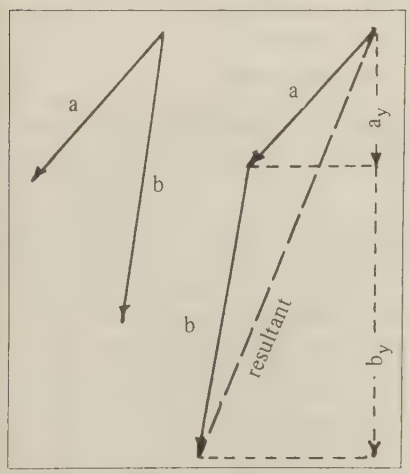


Figure 5. The addition of two forces.

assumed that the planet had a mass very much less than that of either sun so that the planet would have no effect on the orbits of the suns. That assumption can be discarded if desired, but in real solar systems it is pretty close to being correct. Given this assumption the two suns attract each other and both of them attract the planet.

The mutual attraction of the suns we have already considered. Let's look at the planet. There are two forces acting on it. One due to sun 1, the other to sun 2. Each of these forces can be resolved into X and Y components and then the two X components can be added together and the two forces in the Y direction can be added together giving a net X and a net Y force. Figure 5 shows this addition of forces graphically.

Let the Sunshine In

One of the important characteristics of a planetary orbit is the amount of energy received from the sun or suns. Even with a single sun this can be a problem. Figure 6 shows a highly elliptical orbit and it is obvious without any mathematics at all that this planet is going to have cold winters and absolutely ferocious summers. Our old Earth's orbit is only slightly elliptical being about two million miles closer to the sun in January than it is in July. With that small difference the tilt of the axis of

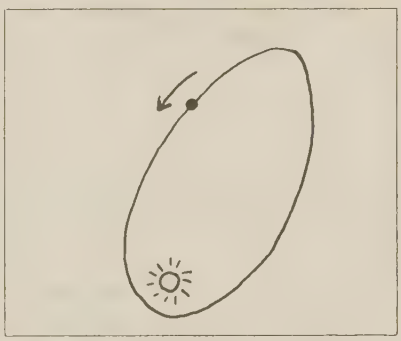


Figure 6. A highly elliptical orbit leading to hot summers and cold winters.

the earth is more important than the increased energy received by the total planet and January turns out to be winter in the northern hemisphere and July is usually considered summer. In the southern hemisphere these two effects reinforce each other rather than opposing as they do up here. Given an equal balance of land and water between the two hemispheres (which we don't have), one would expect that the climate of the southern hemisphere would be more extreme than in the north—the summers hotter and the winters colder. Fortunately, the southern hemisphere is (at present anyway) mostly water covered and the high thermal inertia of the ocean smoothes out this variation.

The axis of the earth being tilted relative to its orbit at 23 1/2 degrees means that the sun is high overhead in summer and low in the sky during winter. On the tropics of Cancer and Capricorn the sun is directly overhead at noon on the first day of summer and only about 45 degrees above the horizon on the first day of winter. It is thermal inertia that causes the hottest and coldest parts of the year to lag behind the days of maximum and minimum solar altitude. If the angle above the horizon is 0 degrees, then the surface of the earth receives an amount of sunshine proportional to the sine of 0; or:

$$S = k \sin \theta$$

Taking the first day of summer as unity, then on the first day of winter the amount of energy received from the sun per square meter is about seven-tenths of unity since the sine of 45 degrees is 0.707.

Our hypothetical planet might have an axial tilt as does the earth but we are going to ignore that for now. However, with two suns of possibly different luminosity (energy output) we are going to have variations in solar input (called "insolation") depending on how close we are to the suns. For the planet to resemble the earth this variation probably should not exceed a ratio of 0.7 to 1 over a year period. Short term variations, during the revolution of one sun about the other, might be larger than this because thermal inertia will tend to smooth them out, but a larger annual variation would make the seasons pretty extreme for comfort.

Figure 7 shows the energy radiated from a star intercepted by a unit area at distance r and that same "cone" of energy at distance 2r. At a distance of 2r the cone intercepts an area of four units—twice as long and twice as high—as it did at distance r. So the energy received per unit area from a

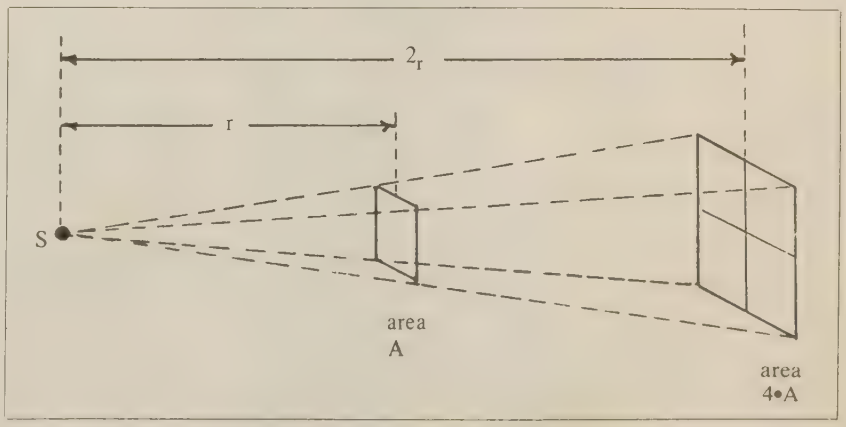


Figure 7. The interception of energy radiated from S.

point source falls off as the square of the distance from the source.

Two suns in our model solar system may have different luminosity. In real life the brightness of a sun is related to its mass and its age, but we are going to ignore that and take any values we like. Again sun 1 will be assumed to be the standard and have luminosity of 1 while sun 2 has luminosity of L2. Then the insolation will be:

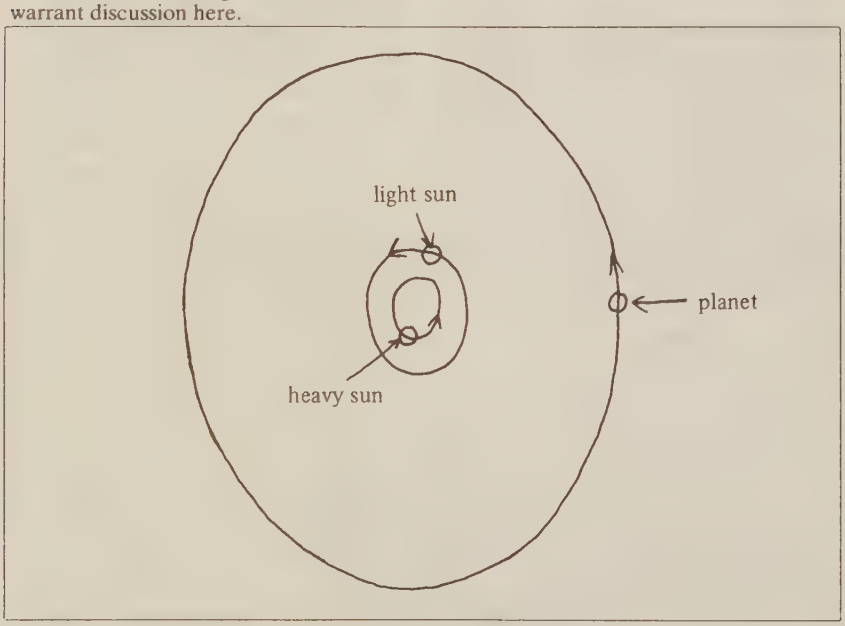
$$I = \frac{1}{R_1^2} + \frac{L2}{R_2^2}$$

where R_1 and R_2 are the distances of the planet from suns 1 and 2 respectively.

Plotting the Results

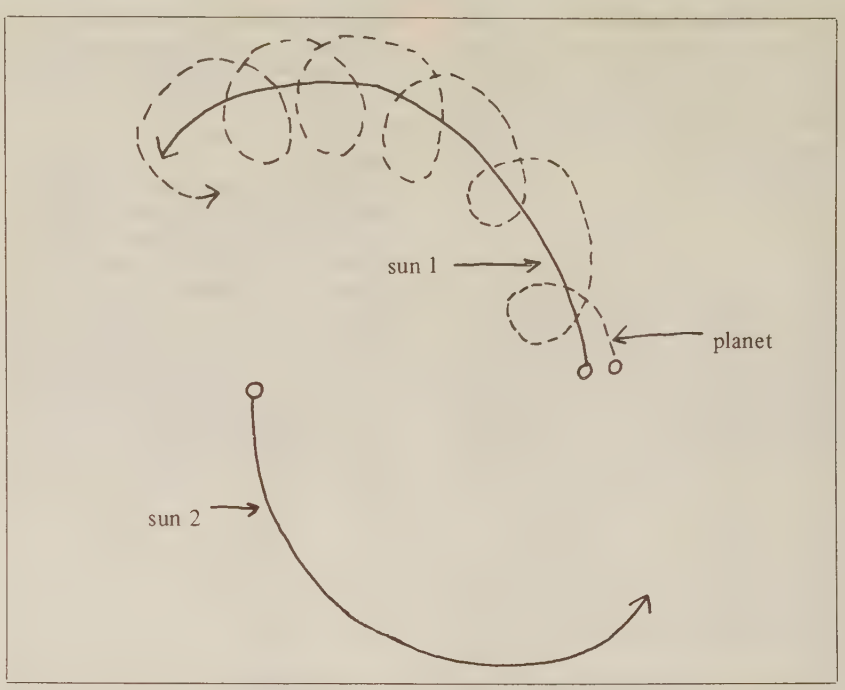
My program was written for the PET which, while it has many interesting characters that can be displayed, is not really intended for plotting continuous curves in space. The smallest unit that can be placed at an arbitrary location on the screen is a quarter square. There are, naturally, four of these characters:

The screen itself is 40 characters wide by 25 high, which gives us 80 units of width and 50 of height. We assume that the center of the screen represents the point (0,0) with positive X to the right and positive Y up. The details of the plot routine are presented in the listing and are not of sufficient general interest to warrant discussion here.



$M_2 = 2$ $X_1 = 5$ $V_1 = .4$
$L_2 = 2$ $X_p = 20$ $V_p = .39$
Max sunshine = .84
Min sunshine = .67

Figure 8. A stable orbit around both suns.



$M_2 = 1$ $X_1 = 15,$ $V_1 = .15$
$L_2 = 1$ $X_p = 20,$ $V_p = .6$
Max shine = 4.52
Min shine = 3.54

Figure 9. A stable or bit around sun 1.

Some Experiments

I'm still exploring various solar systems whenever I can get time on "my" machine. I'll report on only three here.

System 1. Stable orbit around both suns. (See Figure 8). In this system sun 2 is twice as heavy and twice as bright as sun 1. The initial position of sun 1 is at point 5.0 with an upward velocity of 0.4. Consequently, sun 2 will be located at point -2.0 moving down at 0.2 in order to keep the center of mass in the center of the screen. The planet is located initially at 20.0 moving upward at 0.39. As may be seen from Figure 8, the two suns revolve about their common center of mass in concentric ellipses and the planet about both suns in a larger ellipse. The maximum insolation (sunshine) is 0.84 and the minimum is 0.67 for a ratio of 0.8 to 1 which is smaller than that observed in our own topics. This then would be an eminently liveable planet.

System 2. Stable orbit about one of the suns. (See Figure 9). Here the two suns are of equal mass and luminosity. They revolve around each other in a pair of interlocking ellipses, while the planet traces out a flattened corkscrew path as it revolves about one of the suns. Initial conditions placed sun 1 at point 15.0 moving up at 0.15 and the planet at 20.0 moving up at 0.6. The ratio of minimum to maximum insolation is 0.78 to 1 which is about the

same as the previous system. No doubt adjustments could be made to the initial conditions that would cause the suns to maintain a constant distance from each other and hence, keep the insolation even more nearly constant.

System 3. An unstable orbit. Here we have one of the many possible unstable situations. Sun 2 is again twice the mass of sun 1 and they dance around each other now approaching and receding. This, however, is not what makes the orbit of the planet unstable. The problem is that the planet starts out with too small a velocity. It spirals down until it closely approaches one of the suns where a "gravity whip" causes it to gain speed and be thrown out of the system. This gravity whip has been used in flybys of Jupiter to accelerate and redirect some of the Pioneer spacecraft on their way to Saturn.

I'm still exploring solar systems. It's as much fun as reading about them. As a matter of fact, maybe one might serve as a good background for a story of my own:

"John stood on the beach watching the blue sun set behind the waves. The red companion glared down from high in the sky. It was the beginning of the little summer when the planet approached the bright blue star most closely; and already it was hot and humid. John heard a noise behind him and turned..." □

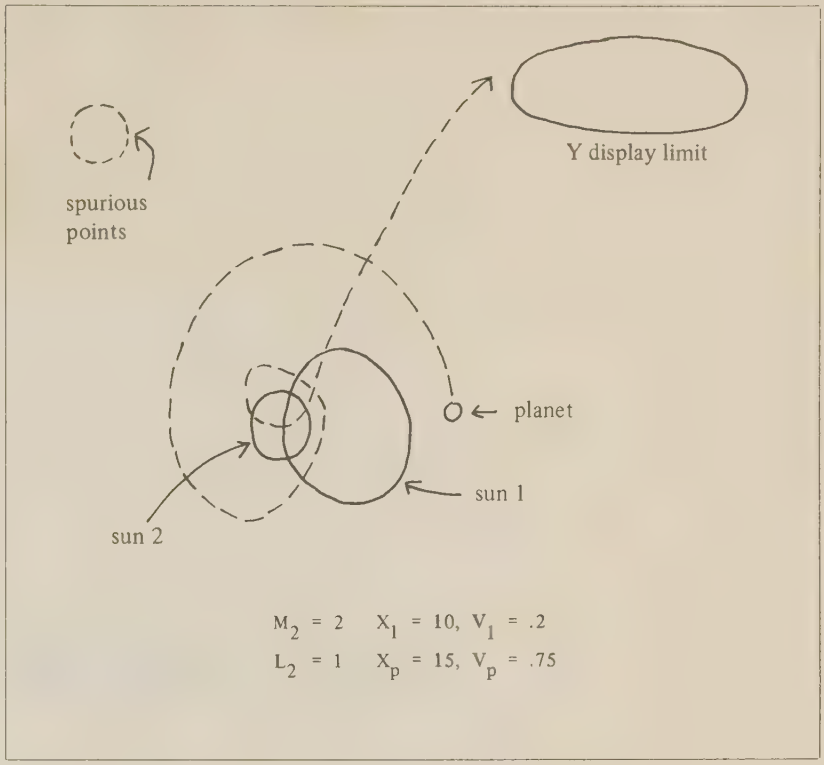


Figure 10. An unstable orbit.

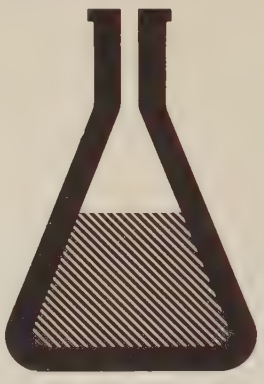
```
5 TS=0:BS=1000
10 REM TWO SUNS ONE PLANET
20 PRINT "INPUT DATA ON SUN 2"
30 PRINT "MASS=";
40 INPUT M2
50 PRINT "LUMINOSITY=";
60 INPUT L2
70 PRINT"INITIAL X OF SUN 1=";
80 INPUT X(1)
90 X(2)=-X(1)/M2
100 PRINT "INITIAL VY OF SUN 1=";
110 INPUT VY(1)
120 VY(2)=-VY(1)/M2
130 PRINT "INITIAL X OF PLANET=";
140 INPUT X(3)
150 PRINT "INITIAL VY OF PLANET=";
160 INPUT VY(3)
170 DT=1
180 PRINT"■"
190 REM RA IS THE DISTANCE BETWEEN THE SUNS
200 RA=((X(1)-X(2))↑2+(Y(1)-Y(2))↑2)↑.5
210 REM RB IS THE DISTANCE FROM PLANET TO SUN 1
220 RB=((X(1)-X(3))↑2+(Y(1)-Y(3))↑2)↑.5
230 REM RC IS THE DISTANCE FROM PLANET TO SUN 2
240 RC=((X(2)-X(3))↑2+(Y(2)-Y(3))↑2)↑.5
250 REM AX IS ACCELERATION IN THE X DIRECTION
260 AX(1)=M2*(X(2)-X(1))/RA↑3
270 AX(2)=-AX(1)/M2
280 AX(3)=(X(1)-X(3))/RB↑3+M2*(X(2)-X(3))/RC↑3
290 REM AY IS ACCELERATION IN Y DIRECTION
300 AY(1)=M2*(Y(2)-Y(1))/RA↑3
310 AY(2)=-AY(1)/M2
320 AY(3)=(Y(1)-Y(3))/RB↑3+M2*(Y(2)-Y(3))/RC↑3
330 FOR I= 1TO 3
340 VX(I)=VX(I)+DT*AX(I)
350 VY(I)=VY(I)+DT*AY(I)
360 X(I)=X(I)+DT*VX(I)
370 Y(I)=Y(I)+DT*VY(I)
380 PX=X(I)
390 PY=Y(I)
400 GOSUB 9000
410 NEXT I
420 REM CALCULATE HEADING INFORMATION
430 T=T+1
440 IF Y(1)<0 THEN F1=1
450 IF Y(1)>0 AND F1=1 THEN M=M+1:F1=0
460 IF Y(3)<0 THEN F2=1
470 IF Y(3)>0 AND F2=1 THEN Y=Y+1:F2=0
480 PRINT"■"
490 PRINT"YEAR=";Y,"SUNTH=";M
500 PRINT "TIME=";T
510 IS=1/RB↑2+L2/RC↑2
520 PRINT "SHINE=";100*IS
530 IFIS>TS THEN TS=IS
540 IF IS <BS THEN BS=IS
550 PRINTTS;BS
560 GOTO 190
9000 REM PLOT THE POINT PX,PY
9010 TX=INT(PX/2)
9020 TY=INT(PY/2)
9030 DX=INT(PX-2*TX)
9040 DY=INT(PY-2*TY)
9050 D=2*DX+DY
9060 DS(0)=123:DS(1)=126:DS(2)=108 DS(3)=124
9070 IF TX>20 THEN TX=20
9080 IF TX<-20 THEN TX=-20
9090 IF TY>12 THEN TY=12
9100 IF TY <-12 THEN TY=-12
9110 REM SHIFT CENTER
9120 TX=TX+20
9130 TY=13-TY
9140 REM CALCULATE POSITION ON SCREEN
9150 L=40*TY+TX+32768
9160 POKE L,DS(D)
9170 RETURN
READY.
```

Chapter Five

Chemistry

Inorganic Chemistry Program

J.P. Peer



A student shows how he used a Sol computer to assist other students in their lab work.

Dear David Ahl:

I read the following interview with Lee Felsenstein on the Sol computer in the July-August 1977 *Creative Computing.*

Ahl: Do you have some feeling concerning what people are doing with them so far?

Felsenstein: It's very hard for me to tell ... they're getting SOLs in order to do something. Just what it is we don't know. ....

In an effort to clear up some of the mystery I am sending you a sample of what I have been doing with my Sol computer. I am a chemistry student and computer fanatic. I bought the Sol because it comes assembled and it has a keyboard instead of those ludicrous toggle switches!

Partial Sample Run

```
YOU CAN PICK THE DISPLAY SPEED FOR THIS PROGRAM.
'0' IS STANDARD (FAST DISPLAY), '99' IS VERY SLOW.
Type the display speed ( 0 to 99 ) here:  0


Type the number of Stability Constants ( 1 to 10 ) here:  5


   OKAY.  NOW TYPE IN THE STABILITY CONSTANTS.
Type K( 1 ) here:  1.9E 4
Type K( 2 ) here:  3.5E 3
Type K( 3 ) here:  7.9E 2
Type K( 4 ) here:  1.5E 2
Type K( 5 ) here:  0.28

NOW YOU NEED TO TYPE IN THE ANALYTICAL CONCENTRATIONS
OF THE LIGAND AND METAL ( PLEASE USE MOLES / LITER ).
Type the Ligand concentration ( 0 to 50 ) here:  1.0
Type the Metal concentration ( 0 to 50 ) here:  0.1


ALPHAS                      CONCENTRATIONS OF COMPLEXES
++++++                      +++++++++++++++++++++++++++
# 0 = 9.8625933 E-14   0-Ligand complex = 9.8625933 E-15 molar
# 1 = 1.8738927 E-9    1-Ligand complex = 1.8738927 E-10 molar
# 2 = .00000656        2-Ligand complex = .00000066 molar
# 3 = .00518131        3-Ligand complex = .00051813 molar
# 4 = .77719701        4-Ligand complex = .0777197 molar
# 5 = .21761516        5-Ligand complex = .02176152 molar
```

Program Listing

```
1000 REM CHM342C MODIFIED FOR SOL.  1977 BY J. P. PEER
1010 REM 417 WEST WATER STREET, BERNE, INDIANA  46711
1020 REM LET'S CLEAR THE CRT SCREEN.  TO DO THIS WE PRINT THE
1030 REM SCREEN-ERASE CHARACTER.  ON SOL IT IS THE 'CLEAR'
1040 REM KEY.  ON OTHER COMPUTERS IT IS THE 'SCREEN ERASE'
1050 REM KEY OR 'CONTROL-L.'  IF YOU ARE USING A HARD-COPY
1060 REM TERMINAL YOU CAN OMIT THIS.
1100 PRINT"(clear key)"
1110 REM SET SPEED OF SOL'S VIDEO DISPLAY MODULE (VDM).  IF
1120 REM YOUR MACHINE IS NOT A SOL YOU CAN OMIT THIS.
1200 SET S=0
1250 DIM K(11),A1(12)
1290 REM A COLON  (:)  SEPARATES STATEMENTS ON ONE LINE.
1300 PRINT:PRINT:PRINT
1400 PRINT"    H I !    T H I S    P R O G R A M    F I N D S    T H E"
1500 PRINT"C O N C E N T R A T I O N S    O F    A L L    S P E C I E S"
1600 PRINT"I N    A    C O M P L E X    I O N    E Q U I L I B R I U M ."
1700 PRINT"Y O U    M U S T    T Y P E    I N    T H E    S T A B I L I T Y"
1800 PRINT"C O N S T A N T S .":PRINT:PRINT:PRINT
1850 PRINT:PRINT:PRINT
1900 REM IF YOU'RE USING A HARD-COPY TERMINAL YOU CAN OMIT THE
1910 REM SUBROUTINE AT LINE 64000.
1920 GOSUB 64000
```

In the Spring of 1976 I began working with Dr. L.A. Bares of Indiana University—Purdue University at Fort Wayne to build a series of interactive programs to assist Inorganic Chemistry students with their laboratory work. This group of programs was written in BASIC on a CDC-6600 and was used this Spring by chemistry students. The series presently has about ten programs. I am sending you one of these programs that I modified to run on my Sol.

This program finds the concentrations of the various species present in a complex-ion equilibrium. The user enters the stability constants and the analytical concentrations of the metal and ligand. The effects of pH and solubility on the equilibrium have been ignored for simplicity.

Our other Inorganic Chemistry programs assist the lab student as he/she finds the stability constants of a complex ion and then, with these constants known, proceed to plot beautiful logarithmic distribution graphs showing the relationship between ligand concentration and various complex ion concentrations.

We also wrote a program to do similar plots of polyprotic acid species versus pH, and a program to plot redox titration curves, just for fun.

The Sol-compatible versions of several of our other programs must wait until Processor Technology produces its long-awaited 8K BASIC. (I've been waiting since February!)

I believe the only non-standard BASIC statement I have used is: SET S = (a number). This changes the video display speed for Processor Technology's VDM. If you aren't using a Processor Technology computer, omit the statements about display speed.

For more information on our programs contact

J. P. Peer
417 West Water Street
Berne, Indiana 46711

and

Dr. L. A. Bares
Indiana University—Purdue University at Fort Wayne
2101 Coliseum Boulevard East
Fort Wayne, Indiana 46805

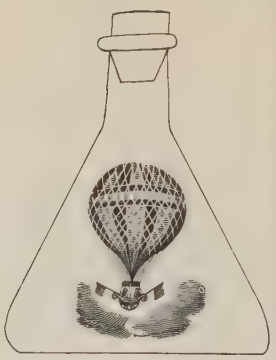
References

Angelici, R. J., *Synthesis & Technique in Inorganic Chemistry, Experiment* 13 (1977).

Blackburn, T. R., *Equilibrium* (1969).

Li, N. C., J. M. White & R. L. Yoest, *J. Amer. Chem. Soc.*, 78, 5218 (1956).

```
2000 FOR Z=1 TO 6:PRINT:NEXT Z
2100 PRINT"          August, 1977 by J. P. Peer"
2200 FOR Z=1 TO 7:PRINT:NEXT Z
2300 GOSUB 64000
2400 PRINT
2500 L=0:U=99
2600 PRINT"YOU CAN PICK THE DISPLAY SPEED FOR THIS PROGRAM."
2700 PRINT"'0' IS STANDARD (FAST DISPLAY), '99' IS VERY SLOW."
2800 INPUT"Type the display speed ( 0 to 99 ) here: ",S2
2900 A=S2
3000 GOSUB 60000
3050 REM CHANGE VDM DISPLAY SPEED TO USER'S WHIM.
3100 SET S=A
3200 PRINT"(clear key)"
3300 L=1:U=10
3400 PRINT:PRINT
3500 INPUT"Type the number of Stability Constants ( 1 to 10 ) here: ",L1
3510 REM IF YOUR BASIC WON'T ACCEPT THIS TYPE OF INPUT STATEMENT TRY
3520 REM THE FOLLOWING SUBSTITUTION:
3530 REM           PRINT"Type the number of ..... here: ";
3540 REM           INPUT A
3600 A=L1
3700 GOSUB 60000
3800 L1=INT(A)
4000 PRINT"(clear key)"
4100 PRINT:PRINT
4200 PRINT"   OKAY.  NOW TYPE IN THE STABILITY CONSTANTS."
4300 FOR I=1 TO L1
4400 PRINT"Type K(";I;" ) here: "; : INPUT" ",K(I)
4500 NEXT I
4600 PRINT
4700 L=0:U=50
4800 PRINT"NOW YOU NEED TO TYPE IN THE ANALYTICAL CONCENTRATIONS"
4900 PRINT"OF THE LIGAND AND METAL ( PLEASE USE MOLES / LITER )."
5000 INPUT"Type the Ligand concentration ( 0 to 50 ) here: ",L2
5100 A=L2
5200 GOSUB 60000
5300 L2=A
5400 INPUT"Type the Metal concentration ( 0 to 50 ) here: ",M
5500 A=M
5600 GOSUB 60000
5700 M=A
5800 REM FIND alpha 0.  SEE BLACKBURN'S 'EQUILIBRIUM' FOR THE THEORY.
5900 T=1:S=1:N=1
6000 T=T*K(N)*L2
6100 S=S+T
6200 N=N+1
6300 IF N <= L1 THEN 6000
6400 A1(0)=1/S
6500 REM FIND ALL THE OTHER alphas.
6600 N=1
6700 A1(N)=K(N)*A1(N-1)*L2
6800 N=N+1
6900 IF N <=L1 THEN 6700
7000 PRINT"(clear key)"
7100 PRINT
7200 PRINT"ALPHAS";TAB(27);"CONCENTRATIONS OF COMPLEXES"
7300 PRINT"++++++";TAB(27);"++++++++++++++++++++++++++++"
7400 FOR J=0 TO L1
7500 PRINT"#";J;" =";A1(J);TAB(21);J;"-Ligand complex =";A1(J)*M;
7550 PRINT" molar"
7600 NEXT J
7650 PRINT
7700 INPUT"To rerun this program type a zero here: ",Z
7800 IF Z ≠ 0 THEN STOP
7900 PRINT"(clear key)"
8000 INPUT"To use the same Stability Constants type zero here: ",Z
8100 IF Z = 0 THEN 4600
8200 GO TO 2400
59999 STOP
60000 REM MY WORLD-FAMOUS ERROR LIMITS SUBROUTINE.  LET'S NOT ALLOW
60010 REM THE USER TO MAKE MISTAKES WHEN S/HE TYPES IN DATA.  INPUT
60020 REM MUST BE WITHIN THE LIMITS OF 'L' AND 'U'.  'A' IS RETURNED.
60030 REM ANY GOOD BASIC INTERPRETER SHOULD HAVE THIS ABILITY BUILT
60040 REM IN AS AN INTRINSIC FUNCTION.
60100 IF A >= L THEN IF A <= U THEN RETURN
60200 PRINT"OOOOPS!  THIS NUMBER MUST BE FROM ";L;" TO ";U
60300 PRINT"PLEASE TRY AGAIN HERE: "; : INPUT" ",A
60400 GO TO 60100
63999 STOP
64000 REM SUBROUTINE TO STOP SCROLLING OF THE SCREEN.  LET'S HAVE
64010 REM PITY ON THE POOR USER WHO CAN'T READ AT 120000 BAUD AND
64020 REM ALLOW HIM/ER TO READ WHAT WE PRINTED BEFORE IT DISAPPEARS.
64030 REM A GOOD INTERPRETER WOULD HAVE THIS AS AN INTRINSIC FUNCTION
64040 REM ALSO.  FOR WORK WITH CRT'S IT'S INEVITABLE.
64100 INPUT"To continue, type any number here: ",Z
64200 PRINT"(clear key)"
64300 RETURN
```



Demonstrating the Gas Laws

David A. Holko

Gas laws frequently present problems for students studying General Chemistry. One reason for the difficulty is the student's inexperience in working with gases in a laboratory.

The concepts involving an ideal gas confined to a container where pressure on the gas, temperature of the gas, or number of moles of gas can be controlled remain nebulous even when examples of tire pumps, aerosol cans, balloons or lab bottles are explained.

This program, when used as a student activity, attempts to demonstrate the relationship between volume and pressure for Boyle's Law, volume and temperature for Charles's Law, or volume and moles of gas for Avagadro's Law.

Initially, the student selects the law which is to be demonstrated. The video monitor shows a simple cylinder, and positions the piston, which traps a volume of gas and is ready to respond to changes in temperature, pressure, or moles of gas confined respective of the law chosen.

When an increase or decrease is entered, the piston moves, correspondingly, showing the effect on the volume of confined gas. The changes are calculated using the ideal gas law $V = nRT/P$ where V is volume, n is the number of moles of gas, R the universal gas constant, T the temperature and P the pressure.

If the student graphs the computer calculated values, checks the values by doing hand calculations, or closely analyzes the values calculated by the computer, the relationship of molar gas volume at Standard Temperature and Pressure may become obvious.

The advanced student, via graphing and analysis, can be led to determine the actual value of R, the Universal Gas constant, by using the computer simulation as an experimental apparatus.

There is a minimum volume the cylinder will hold. This design was effected to allow graphic extrapolation of the data to determine the value of Absolute Zero with Charles's Law.

The programming is simple and short, written to aid and reinforce the student's application of the gas laws and equate a simulated moving piston to theoretical values derived using the ideal gas law. □

```
  1 'GAS LAWS VERSION 3.0  WRITTEN BY DAVID A. HOLKO
2 CLS:
  B$="###.###":
  PP=1:
  P=.082056:
  N=1:
  T=273.15:
  N(1)=1:
  N(2)=40:
  N(3)=1:
  U$(1)="GAS PRESSURE":
  U$(2)="GAS TEMPERATURE":
  U$(3)="MOLES OF GAS":
  F(1)=.01:
  F(2)=9:
  F(3)=.025
3 PRINT"1 BOYLES LAW":
  PRINT"2 CHARLES LAW":
  PRINT"3 AVOGADRO'S LAW":
  INPUT"CHOOSE FROM THE ABOVE LAWS";C:
  IFC>3ORC<1THENCLS:
  GOTO3    ELSECLS
4 PRINT@70,"GAS CYLINDER WITH PISTON  ";:
  PRINTCHR$(92);:
5 FORX=1TO120:
  SET(X,8):
  SET(X,14):
  NEXTX:
  FORX=9TO14:
  SET(1,X):
  SET(2,X):
  NEXTX
```

David A. Holko, 642 N. River Dr. #102, Stuart, FL 33494.

```
 6 IFC=1THENPRINT@778,"PRESSURE    ";:
   PRINTUSINGB$;N(C);:
   PRINT"  ATM";
 7 PRINT@714,"VOLUME       ";:
   PRINTUSINGB$;V;:
   PRINT"  LITERS";
 8 IFC=2THENPRINT@778,"TEMPERATURE ";:
   PRINTUSINGB$;N(C);:
   PRINT"  DEGREES K";
 9 IFC=3THENPRINT@778,"MOLES OF GAS";:
   PRINTUSINGB$;N(C);:
   PRINT"  MOLES";
10 PRINT@900,"TYPE (I) TO INCREASE OR (D) TO DECREASE THE ";V$
   (C);".";
11 R$(1)="THE PRESSURE ON THE GAS IS APPLIED BY THE PISTON."
12 R$(3)="THE MASS OF GAS IN THE CYLINDER CAN BE CHANGED."
13 H$(3)="PRESSURE & TEMPERATURE ARE HELD CONSTANT AT STP."
14 R$(2)="THE TEMPERATURE OF THE CYLINDER HOLDING THE GAS CAN
   BE CHANGED."
15 H$(2)="ONE MOLE OF GAS & STD. PRESSURE ARE HELD CONSTANT."
16 H$(1)="ONE MOLE OF GAS & STD. TEMPERATURE ARE HELD CONSTANT
   ."
17 PRINT@448,R$(C);:
   PRINT@512,H$(C);
   IFZZ=0THENGOTO21
18 PRINT@980,"TYPE (R) TO RERUN PROGRAM.";
19 A$=INKEY$:
   IFLEN(A$)=0:
   GOTO19
20 IFA$="I"THENN(C)=N(C)+F(C)ELSEIFA$="D"THENN(C)=N(C)-F(C)ELS
   EIFA$="R"THENRUNELSEGOTO19
21 IFC=1THENV=(N*R*T)/N(1)
22 IFC=2THENV=(N*R*N(C))/PP
23 IFC=3THENV=(N(C)*R*T)/PP
24 ZZ=1:
   P=V+200:
   IFP<201THENGOTO26   ELSEIFP>245THENGOTO27   ELSEGOSUB25   :
   GOTO6
25 PRINT@P,CHR$(191);:
   PRINT@P+64,CHR$(191);:
   PRINT@P-1,CHR$(128);:
   PRINT@P+63,CHR$(176);:
   PRINT@P+1,CHR$(128);:
   PRINT@P+65,CHR$(176);:
   PRINT@P+1,CHR$(176);CHR$(93)" PISTON  ";:
   PRINT@P+120,"  ";CHR$(91);" TRAPPED GAS ";:
   RETURN
26 IFC=3ORC=2THENPRINT@512,"THE MINIMUM VOLUME OF THIS CYLINDE
   R HAS BEEN REACHED.";:
   GOTO19   ELSEPRINT@512,"THE MAXIMUM PRESSURE THIS PISTON CA
   N EXERT HAS BEEN REACHED.";:
   GOTO19
27 PRINT@512,"THE MAXIMUM VOLUME FOR THIS CYLINDER HAS BEEN RE
   ACHED.";:
   GOTO19
```

```
      GAS CYLINDER WITH PISTON  \
##############################################################
##                              ###] PISTON
##############################################################
                        [ TRAPPED GAS

THE PRESSURE ON THE GAS IS APPLIED BY THE PISTON.
ONE MOLE OF GAS & STD. TEMPERATURE ARE HELD CONSTANT.


          VOLUME        22.414  LITERS
          PRESSURE       1.000  ATM

    TYPE (I) TO INCREASE OR (D) TO DECREASE THE GAS PRESSURE.
                   TYPE (R) TO RERUN PROGRAM.


                    BOYLE'S LAW SCREEN
```

```
      GAS CYLINDER WITH PISTON  \
##################################################################
##         ###] PISTON
##################################################################
     [ TRAPPED GAS

THE TEMPERATURE OF THE CYLINDER HOLDING THE GAS CAN BE CHANGED.
ONE MOLE OF GAS & STD. PRESSURE ARE HELD CONSTANT.


          VOLUME         3.282  LITERS
          TEMPERATURE   40.000  DEGREES K

    TYPE (I) TO INCREASE OR (D) TO DECREASE THE GAS TEMPERATURE.
                    TYPE (R) TO RERUN PROGRAM.


                    CHARLES' LAW SCREEN


      GAS CYLINDER WITH PISTON  \
##################################################################
##                            ###] PISTON
##################################################################
                          [ TRAPPED GAS

THE MASS OF GAS IN THE CYLINDER CAN BE CHANGED.
PRESSURE & TEMPERATURE ARE HELD CONSTANT AT STP.


          VOLUME        22.414  LITERS
          MOLES OF GAS   1.000  MOLES

    TYPE (I) TO INCREASE OR (D) TO DECREASE THE MOLES OF GAS.
                    TYPE (R) TO RERUN PROGRAM.


                    AVAGADRO'S LAW SCREEN
```

Periodic Table

Dean Schmidt

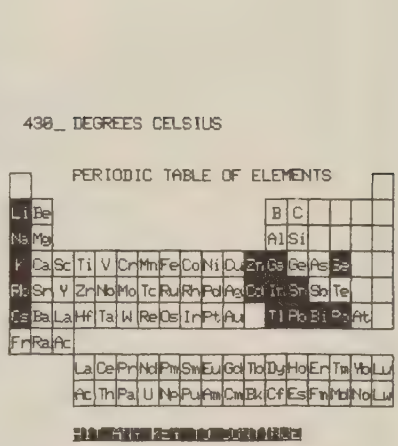
Periodic Table is a program which produces a Periodic Table on the screen with each element symbol in its correct place. Three options are given. The first allows the user to select a group of elements, for example, the transition elements, which are shown in inverse on the periodic table. There are eleven different groups that can be selected.

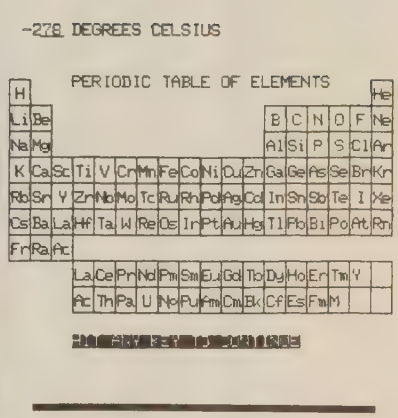
Another option starts with the temperature in the upper left hand starting at absolute zero. As the temperature rises, each element changes to inverse on the screen as it reaches its melting point. When an element reaches its vaporization point its symbol is removed from the periodic table.

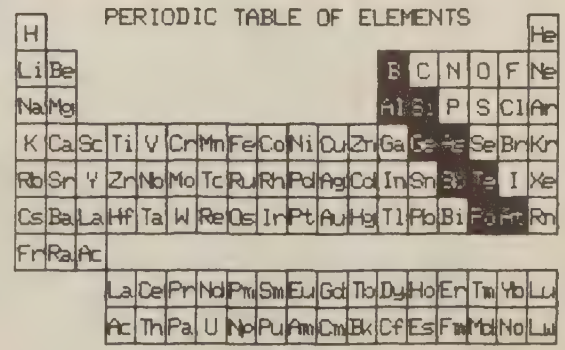
The third option is a variation of the second. The user may select a specific temperature and have the conditions shown for that temperature. For example, if 0° Celcius were chosen, each element that is a solid at that temperature is shown on the screen in normal characters. Each element that is a liquid is shown in inverse, and each element that is a gas is eliminated from the table.

The program is written in Pascal for the Apple II Plus with language card.

Sample runs of Periodic Table are shown here. If you would like a listing of the program, please send $1.00 with an SASE to: Magic City Campus, Minot H.S., Minot, ND 58701, Attn: Computer Club. For a disk containing the program, send $10 or $5 and a blank disk to the same address. □







Dean Schmidt, #1 Southwest 22nd St., Minot, ND 58701.

SYMBOL	MELT	BOIL	X	Y	OVER
H	-259	-253	5	140	0
He	-272	-269	260	140	1
Li	179	1317	5	125	1
Be	1278	2970	20	125	1
B	2300	2550	185	125	0
C	3550	4827	200	125	0
N	-210	-196	215	125	0
O	-218	-183	230	125	0
F	-220	-188	245	125	0
Ne	-249	-246	260	125	1
Na	98	892	5	110	1
Mg	651	1107	20	110	1
Al	660	2467	185	110	1
Si	1410	2355	200	110	1
P	44	280	215	110	0
S	113	45	230	110	0
Cl	-101	-35	245	110	1
Ar	-189	-186	260	110	0
K	64	774	5	95	0
Ca	845	1487	20	95	1
Sc	1539	2727	35	95	1
Ti	1668	3260	50	95	1
V	1890	3000	65	95	0
Cr	1890	2482	80	95	0
Mn	1244	2097	95	95	1
Fe	1535	3000	110	95	1
Co	1495	2900	125	95	1
Ni	1453	2732	140	95	1
Cu	1083	2595	155	95	1
Zn	419	907	170	95	1
Ga	30	2403	185	95	1
Ge	937	2830	200	95	1
As	817	613	215	95	1
Se	217	684	230	95	1
Br	-7	59	245	95	0
Kr	-157	-152	260	95	0
Rb	39	688	5	80	1
Sr	769	1384	20	80	0
Y	1490	2927	35	80	0
Zr	1852	3578	50	80	0
Nb	2468	4927	65	80	1
Mo	2610	5560	80	80	1
Tc	2140	9999	95	80	1
Ru	2450	4111	110	80	1
Rh	1966	3960	125	80	1
Pd	1549	2927	140	80	1
As	961	2212	155	80	1
Cd	321	765	170	80	1
In	157	2000	185	80	1
Sn	232	2270	200	80	1
Sb	631	1380	215	80	1
Te	450	990	230	80	1
I	114	184	245	80	0
Xe	-112	-107	260	80	1
Cs	29	690	5	65	1
Ba	725	1140	20	65	1
La	920	3469	35	65	1
Hf	2150	5400	50	65	1
Ta	2996	5425	65	65	1
W	3370	5900	80	65	0
Re	3167	5627	95	65	1
Os	3050	5000	110	65	1
Ir	2443	4500	125	65	0
Pt	1774	3800	140	65	1
Au	1063	2966	155	65	1
Hg	-39	366	170	65	1
Tl	304	1457	185	65	1
Pb	327	1744	200	65	1
Bi	271	1560	215	65	1
Po	254	962	230	65	1
At	9999	9999	245	65	1
Rn	-71	-62	260	65	1
Fr	9999	9999	5	50	0
Ra	700	1140	20	50	1
Ac	1050	3200	35	50	1
La	920	3469	50	35	1
Ce	795	3468	65	35	1
Pr	935	3127	80	35	0
Nd	1024	3027	95	35	1
Pm	1035	2730	110	35	1
Sm	1072	1900	125	35	1
Eu	826	1439	140	35	1
Gd	1312	3000	155	35	1
Tb	1356	2530	170	35	1
Dy	1407	2600	185	35	1
Ho	1461	2600	200	35	1
Er	1497	2900	215	35	0
Tm	1545	1727	230	35	1
Yb	824	1427	245	35	1
Lu	1652	3327	260	35	1
Ac	1050	3200	50	20	1
Th	1750	3600	65	20	1
Pa	1230	9999	80	20	1
U	1132	3818	95	20	0
Np	637	9999	110	20	1
Pu	640	3235	125	20	1
Am	995	2607	140	20	1
Cm	9999	9999	155	20	1
Bk	9999	9999	170	20	1
Cf	9999	9999	185	20	1
Es	9999	9999	200	20	1
Fm	9999	9999	215	20	1
Md	9999	9999	230	20	1
No	9999	9999	245	20	1
Lw	9999	9999	260	20	1

Sample runs and the program listings follow. This program is also available on disk for $10 from: Computer Math Department, Magic City Campus, Minot, ND 58701.

```
PROGRAM WRITE;

TYPE DATA=RECORD
            SYMBOL:STRING[2];
            MELT:INTEGER;
            BOIL:INTEGER;
            LOCX:INTEGER;
            LOCY:INTEGER;
            OVER:INTEGER;
          END;

VAR STORE:FILE OF DATA;
    ELEMENT:ARRAY[1..105] OF DATA;
    I,O,X:INTEGER;
    AN,T:STRING;
    CH:CHAR;

PROCEDURE CONVERT;
(**************************)
(* THIS PROCEDURE CHANGES *)
(* THE SECOND CHAR. OF    *)
(* STRING 'SYMBOL' TO     *)
(* LOWER CASE, IF LOWER   *)
(* CASE LETTERS ARE       *)
(* AVAILABLE THROUGH      *)
(* KEYBOARD, REMOVE THIS  *)
(* PROCEDURE              *)
(**************************)

BEGIN
     WITH ELEMENT[I] DO
       BEGIN
        CH:=SYMBOL[2];
        O:=ORD(CH);
        CH:=CHR(O+32);
        SYMBOL[2]:=CH;
       END;
   END;

PROCEDURE SAVE;
(******************************)
(* THIS PROCEDURE STORES THE *)
(* ENTERED DATA ON THE DISK  *)
(******************************)

   BEGIN
      STORE↑:=ELEMENT[I];
      PUT(STORE);
   END;

PROCEDURE OLD;
(**********************************)
(* THIS PROCEDURE FINDS THE       *)
(* END OF A PREVIOUSLY STARTED    *)
(* FILE AND STORES THE LAST       *)
(* RECORD # AS VARIABLE 'X'       *)
(**********************************)

  BEGIN
    RESET(STORE,'PD2');
    I:=0;

    WHILE NOT EOF(STORE) DO
       BEGIN
          I:=I+1;
          ELEMENT[I]:=STORE↑;
          GET(STORE);
       END;
    X:=I;
  END;
```

```
        PROCEDURE NEW;
        (*********************************)
        (* THIS PROCEDURE OPENS A NEW FILE *)
        (*********************************)

  BEGIN
    REWRITE(STORE,'FID2 ');
    X:=1;
  END;


BEGIN (* MAIN PROGRAM *)

  WRITELN('O)LD OR N)EW FILE?');
  READLN(T);
  IF T[1]='O'THEN OLD
    ELSE NEW;

    FOR I:=X TO 105 DO
      WITH ELEMENT[I] DO
        BEGIN
          WRITELN('INPUT SYMBOL');
          READLN(SYMBOL);
          IF SYMBOL='Q' THEN   (* ENTER 'Q' FOR SYMBOL TO END DATA ENTRY*)
            BEGIN
              CLOSE(STORE,LOCK);
              I:=105;
            END  (* ADD ';' IF CONVERT IS REMOVED *)


          (*****************************************)
          (* REMOVE UNTIL NEXT SPACE IF LOWER CASE *)
          (* IS AVAILABLE FROM KEYBOARD            *)
          (*****************************************)
          ELSE
            BEGIN
             IF LENGTH(SYMBOL)>1 THEN CONVERT;

        WRITELN('INPUT MELTING PT');
        READLN(MELT);
        WRITELN('INPUT BOILING PT');
        READLN(BOIL);
        WRITELN('X COORDINATE');
        READLN(LOCX);
        WRITELN('Y COORDINATE');
        READLN(LOCY);
        WRITELN('INPUT OVERLAP');
        READLN(OVER);

        PAGE(OUTPUT);
        WRITELN(' ');
        WRITELN('SYMBOL: ',SYMBOL);
        WRITELN('MELT: ',MELT);
        WRITELN('BOIL: ',BOIL);
        WRITELN('LOCX: ',LOCX);
        WRITELN('LOCY: ',LOCY);
        WRITELN('OVERLAP: ',OVER);
        WRITELN(' ');
        WRITELN('IS DATA CORRECT');
        READLN(AN);

        WRITELN(' ');
        WRITELN(' ');
        IF AN[1]<>'Y' THEN
          BEGIN
            I:=I-1;
            WRITELN('RETYPE ENTIRE ELEMENT');
          END

        ELSE SAVE;(*DATA CORRECT*)
        PAGE(OUTPUT);
      END;
    END;(*END OF WITH*)

  CLOSE(STORE,LOCK);
END.
```

```
(*$S+*)
PROGRAM PERIODIC TABLE;

(*****************************)
(*                           *)
(* WRITTEN BY DEAN SCHMIDT   *)
(* #1-SW-22ND STREET         *)
(* MINOT, ND 58701           *)
(*                           *)
(*****************************)

USES TURTLEGRAPHICS,APPLESTUFF;

TYPE  DATA=RECORD
            SYMBOL:STRING[2];   (* SYMBOL OF ELEMENT          *)
            MELT:INTEGER;       (* MELTING PT OF ELEMENT      *)
            BOIL:INTEGER;       (* BOILING PT OF ELEMENT      *)
            LOCX:INTEGER;       (* X COORDINATE OF BOX        *)
            LOCY:INTEGER;       (* Y COORDINATE OF BOX        *)
            OVER:INTEGER;       (* OVER LAP OF SECOND         *)
                                (* CHAR ON FIRST              *)
          END;(* OF RECORD *)

VAR X,Y:INTEGER;                (* LOCATION OF ELEMENT        *)
    I  :INTEGER;                (* COUNTER                    *)
    T  :INTEGER;                (* COUNTER OF TEMPERATURE     *)
    C  :INTEGER;                (* KEEPS TRACK OF CHOICE      *)
    P,TS:STRING[50];            (* UTILITY VARIABLES          *)
    ELEMENT:ARRAY[1..105] OF DATA;(* STORES CONTENTS OF RECORDS *)



PROCEDURE V;
(*****************************************)
(*   THIS PROCEDURE REVERSES COLORS IN   *)
(*   APPROPRIATE BOX DUE TO MELTING OR   *)
(*   GROUP INCLUSION.                    *)
(*****************************************)

  BEGIN
   WITH ELEMENT[I] DO
    BEGIN
     X:=LOCX+1;
     Y:=LOCY+1;
      VIEWPORT(X,X+13,Y,Y+13);
      FILLSCREEN(REVERSE);
      VIEWPORT(0,279,0,191);
    END;(* OF WITH *)
  END;


PROCEDURE B;
(*****************************************)
(*  THIS PROCEDURE BLACKENS APPROPRIATE   *)
(*  BOX DUE TO REACHING OF BOILING POINT  *)
(*****************************************)

  BEGIN
   WITH ELEMENT[I] DO
    BEGIN
      X:=LOCX+1;
      Y:=LOCY+1;
      VIEWPORT(X,X+13,Y,Y+13);
      FILLSCREEN(BLACK);
      VIEWPORT(0,279,0,191);
    END;(* OF WITH *)
  END;


PROCEDURE SETUP;
(*************************************)
(*  THIS PROCEDURE CREATES THE HI-RES *)
(*  PERIODIC TABLE OF ELEMENTS       *)
(*************************************)


    PROCEDURE PLACE;
    (*********************************)
    (*  THIS PROCEDURE PLACES TWO-CHAR  *)
    (*  LONG SYMBOLS ON PERIODIC TABLE  *)
```

```
  (************************************)
   VAR S,S2:CHAR;(* INDIVIDUAL CHAR OF STRING 'SYMBOL' *)

    BEGIN
     WITH ELEMENT[I] DO
      BEGIN
       Y:=LOCY+3;
       X:=LOCX+2;;
       S:=SYMBOL[1];
       S2:=SYMBOL[2];
       PENCOLOR(NONE);
       MOVETO(X,Y);
       PENCOLOR(WHITE);
       WCHAR(S);
       X:=TURTLEX-OVER;
       PENCOLOR(NONE);
       MOVETO(X,Y);
       PENCOLOR(WHITE);
       CHARTYPE(6);
       WCHAR(S2);
       CHARTYPE(10);
     END;(* OF WITH *)
  END;

 PROCEDURE PLACE2;
 (****************************************)
 (*  THIS PROCEDURE PLACES ONE-CHARACTER *)
 (*  LONG SYMBOLS ON THE PERIODIC TABLE  *)
 (****************************************)

    BEGIN
      WITH ELEMENT[I] DO
       BEGIN
        Y:=LOCY +3;
        X:=LOCX+5;
        PENCOLOR(NONE);
        MOVETO(X,Y);
        PENCOLOR(WHITE);
        WSTRING(SYMBOL);
       END;(* OF WITH *)
  END;

BEGIN(* MAIN PROCEDURE SETUP *)
  INITTURTLE;
  GRAFMODE;

  (* DRAW VERTICAL LINES OF TABLE *)
  PENCOLOR(NONE);
  MOVETO(-10,155);
  REPEAT
   PENCOLOR(NONE);
   X:=TURTLEX;
   MOVETO(X+15,155);
   PENCOLOR(WHITE);
   MOVETO(X+15,20);
  UNTIL TURTLEX>=275;

  (* DRAW HORIZONTAL LINES OF TABLE *)
  PENCOLOR(NONE);
  MOVETO(275,5);
  REPEAT
    PENCOLOR(NONE);
    Y:=TURTLEY;
    MOVETO(275,Y+15);
    PENCOLOR(WHITE);
    MOVETO(5,Y+15);
  UNTIL TURTLEY=155;

  (* REMOVE UNNECESSARY LINES *)
  VIEWPORT(21,259,141,160);
  FILLSCREEN(BLACK);
  VIEWPORT(36,184,111,140);
  FILLSCREEN(BLACK);
  VIEWPORT(51,279,51,64);
  FILLSCREEN(BLACK);
  VIEWPORT(0,49,19,49);
  FILLSCREEN(BLACK);
  VIEWPORT(0,279,0,191);

  PENCOLOR(NONE);
  MOVETO(50,150);
  P:='PERIODIC TABLE OF ELEMENTS';
  PENCOLOR(WHITE);
  WSTRING(P);

  (* PLACE SYMBOLS ON TABLE *)
  FOR I:=1 TO 105 DO
    BEGIN
     WITH ELEMENT[I] DO
       BEGIN
        IF LENGTH(SYMBOL)>1 THEN PLACE
        ELSE PLACE2;
       END;(* OF WITH *)
    END;  (* OF FOR  *)

  (* PLACE INSTRUCTIONS AT BOTTOM *)
  IF C=2 THEN
   BEGIN
    PENCOLOR(NONE);
    MOVETO(50,0);
    P:='HIT ANY KEY TO EXIT';
    PENCOLOR(WHITE);
    CHARTYPE(5);
    WSTRING(P);
    CHARTYPE(10);
   END(* OF IF *)
  ELSE
   BEGIN
    TEXTMODE;
    PENCOLOR(NONE);
    MOVETO(50,0);
    P:='HIT ANY KEY TO CONTINUE';
    PENCOLOR(WHITE);
    CHARTYPE(5);
    WSTRING(P);
    CHARTYPE(10);
   END;(* OF ELSE *)

END;


PROCEDURE TEMP;
(******************************)
(* THIS PROCEDURE SHOWS THE *)
(* EFFECT OF TEMPERATURE    *)
(* INCREASES ON THE STATE   *)
(* OF THE VARIOUS ELEMENTS  *)
(******************************)

 VAR INC:INTEGER;(* INCREMENT VALUE TO SPEED UP PROCEDURE *)

  BEGIN
   SETUP;
   T:=-302;
   INC:=2;
   PENCOLOR(NONE);
   MOVETO(50,180);
   P:='DEGREES CELSIUS';
   PENCOLOR(WHITE);
   WSTRING(P);

  REPEAT
    BEGIN
      T:=T+INC;

      (* WRITE TEMP. ON SCREEN *)
      STR(T,TS);
      PENCOLOR(NONE);
      MOVETO(15,180);
      PENCOLOR(WHITE);
      WSTRING(TS);
```

```
    (* CHECK FOR PHASE CHANGE *)
    FOR I:=1 TO 105 DO
     BEGIN
        WITH ELEMENT[I] DO
        BEGIN
         IF (MELT>T-INC) AND (MELT<=T) AND (MELT<=BOIL) THEN V;
         IF (BOIL>T-INC) AND (BOIL<=T) THEN B;
        END;(* OF WITH *)
     END;     (* OF FOR  *)

     (* REMOVE OLD TEMP. FROM SCREEN *)
     MOVETO(15,180);
     PENCOLOR(WHITE);
     CHARTYPE(6);
     WSTRING(TS);
     CHARTYPE(10);

     (* CHANGE INCREMENTAL VALUE *)
     IF T=0 THEN INC:=5;
     IF T>=500 THEN INC:=10;

  END;
  UNTIL (T>=6000) OR (KEYPRESS);
  UNITCLEAR(1);                  (* CLEARS KEYPRESS *)
  FOR I:=1 TO 3000 DO;           (* DELAY SEQUENCE  *)
END;

PROCEDURE STATUS;
(*****************************)
(* THIS PROCEDURE SHOWS THE *)
(* STATE OF EACH ELEMENT AT *)
(* A TEMPERATURE CHOSEN BY  *)
(* THE USER                 *)
(*****************************)

 VAR T2:INTEGER; (* USED TO COMPARE NEW TEMP *)
                 (* CHOICE WITH LAST CHOICE  *)

  BEGIN
    T2:=-301;
    PAGE(OUTPUT);(* CLEARS TEXT SCREEN *)
    SETUP;
    WRITELN('IF YOU PLAN ON REPEATING THIS PORTION');
    WRITELN('OF THE PROGRAM, START WITH THE LOWEST');
    WRITELN('DESIRED VALUE.  TEMPERATURES MAY RANGE');
    WRITELN('FROM -300 TO 6000 DEGREES CELSIUS. TO');
    WRITELN('EXIT THIS PART OF THE PROGRAM JUST');
    WRITELN('ENTER A NUMBER OUT OF THIS RANGE');
    WRITELN('   ');
    WRITELN('   ');
  REPEAT
   REPEAT
    TEXTMODE;
    GOTOXY(1,8);
    WRITELN('      ENTER TEMPERATURE                ');
    GOTOXY(24,8);
    READ(T);
    IF (T<-300) OR (T>6000) THEN EXIT(STATUS);
   UNTIL T>=T2;

    (* PLACE TEMP. ON SCREEN *)
    STR(T,TS);
    P:=' DEGREES CELSIUS';
    P:=CONCAT(TS,P);
    PENCOLOR(NONE);
    MOVETO (15,180);
    PENCOLOR(WHITE);
    WSTRING(P);
    GRAFMODE;

    (* CHECK STATE OF EACH ELEMENT *)
    FOR I:=1 TO 105 DO
      BEGIN
       WITH ELEMENT[I] DO
        BEGIN
         IF (MELT<=T) AND (MELT>T2) AND (MELT<=BOIL) THEN V;
         IF (BOIL<=T) AND (BOIL>T2) THEN B;
        END;(* OF WITH *)
      END;  (* OF FOR  *)

    REPEAT
    UNTIL KEYPRESS;
    UNITCLEAR(1);  (* CLEARS KEYPRESS *)
    TEXTMODE;
    T2:=T;

    (* ERASE OLD TEMP. *)
    PENCOLOR(NONE);
    MOVETO(15,180);
    CHARTYPE(6);
    WSTRING(P);
    CHARTYPE(10);

 UNTIL (T<-300) OR (T>6000);
END;


PROCEDURE GROUP;
(*********************************)
(* THIS PROCEDURE LOCATES VARIOUS *)
(* GROUPS OF ELEMENTS ON THE      *)
(* PERIODIC TABLE                 *)
(*********************************)

 VAR ES,M,BR,LO,DC,HL,LM,NM,RG,TR,LA,AC: SET OF 1..105;(* IDENTIFY GROUPS *)

   PROCEDURE ASSIGN;
   (*************************************)
   (* THIS PROCEDURE ASSIGN VALUES TO *)
   (* THE VARIOUS SET VARIABLES       *)
   (*************************************)

     BEGIN
       LA:=[76..90];
       AC:=[91..105];
       RG:=[2,10,18,36,54,72];
       TR:=[5,13,14,32,33,51,52,70,71];
       LM:=[3,4,11,12,19,20,37,38,55,56,73,74];
       NM:=[5,..9,14,..17,33,..35,52,53,71];
       BR:=[21..25,39..41];
       LO:=[13,30..32,48,..51,66,..70];
       DC:=[26..29,42,..47,57..65,75];
       HL:=[9,17,35,53,71];

     END;

   PROCEDURE DUCT;
   (*********************************)
   (* THIS PROCEDURE LOCATES THE  *)
   (* DUCTILE METALS ON THE TABLE *)
   (*********************************)

     BEGIN
       GRAFMODE;

       (* PLACE GROUP NAME ON SCREEN *)
       PENCOLOR(NONE);
       MOVETO(50,175);
       PENCOLOR(WHITE);
       CHARTYPE(5);
       WSTRING(P);
       CHARTYPE(6);

      (* CHECK FOR GROUP INCLUSION *)
      FOR I:=1 TO 105 DO
       BEGIN
        WITH ELEMENT[I] DO
         BEGIN
```

```
      IF (I IN DC) OR (I IN LA) OR (I IN AC) THEN V;
    END;(* OF WITH *)
  END;  (* OF FOR  *)

  REPEAT
  UNTIL KEYPRESS;
  UNITCLEAR(1);  (* CLEARS KEYPRESS *)

  (* RETURN TABLE TO NORMAL *)
  PENCOLOR(NONE);
  MOVETO(50,175);
  PENCOLOR(WHITE);
  CHARTYPE(9);
  WSTRING(P);
  CHARTYPE(6);
 FOR I:=1 TO 105 DO
   BEGIN
     WITH ELEMENT[I] DO
       IF (I IN DC) OR (I IN LA) OR (I IN AC) THEN V;
   END;(* OF FOR *)

END;

PROCEDURE METALS;
(*******************************)
(* THIS PROCEDURE LOCATES THE *)
(* REMAINDER OF THE METALS    *)
(*******************************)

 BEGIN
   GRAFMODE;

   (* WRITE GROUP NAME ON SCREEN *)
   PENCOLOR(NONE);
   MOVETO(50,175);
   PENCOLOR(WHITE);
   CHARTYPE(5);
   WSTRING(P);
   CHARTYPE(6);

   (* CHECK FOR GROUP INCLUSION *)
   FOR I:=1 TO 105 DO
     BEGIN
       WITH ELEMENT[I] DO
         BEGIN
           IF (I IN BR) OR (I IN LO) THEN V;
         END;(* OF WITH *)
     END;    (* OF FOR  *)

END;

PROCEDURE RMET;
(*********************************)
(* THIS PROCEDURE REVERSES THE   *)
(* EFFECT OF PROCEDURE 'METALS'  *)
(* TO PREVENT THE NEED FOR A NEW *)
(* TABLE TO BE CONSTRUCTED       *)
(*********************************)

 BEGIN
  FOR I:=1 TO 105 DO
   BEGIN
    WITH ELEMENT[I] DO
     IF (I IN BR) OR (I IN LO) THEN V;
   END;(* OF FOR *)
 END;

PROCEDURE WORK;
(*********************************)
(* THIS PROCEDURE LOCATES ALL    *)
(* GROUPS EXCEPT FOR THE METALS  *)
(* AND DUCTILE METALS            *)
(*********************************)

  BEGIN
    GRAFMODE;

    (* PLACE GROUP NAME ON SCREEN *)
    PENCOLOR(NONE);
    MOVETO(50,175);
    PENCOLOR(WHITE);
    CHARTYPE(5);
    WSTRING(P);
    CHARTYPE(6);

    (* CHECK FOR GROUP INCLUSION *)
    FOR I:=1 TO 105 DO
     BEGIN
      WITH ELEMENT[I] DO
        BEGIN
          IF I IN ES THEN V;
        END;(* OF WITH *)
     END;    (* OF FOR  *)

    REPEAT
    UNTIL KEYPRESS;
    UNITCLEAR(1);  (*CLEARS KEYPRESS *)

    (* RETURN TABLE TO NORMAL *)
    PENCOLOR(NONE);
    MOVETO(50,175);
    PENCOLOR(WHITE);
    CHARTYPE(9);
    WSTRING(P);
    CHARTYPE(6);
    FOR I:=1 TO 105 DO
      BEGIN
        WITH ELEMENT[I] DO
        IF I IN ES THEN V;
      END;(* OF FOR *)
  END;

BEGIN(* MAIN PROCEDURE 'GROUP' *)

 PAGE(OUTPUT);(* CLEARS TEXT SCREEN *)
 ASSIGN;
 SETUP;
 TEXTMODE;

 (* PRINT CHOICES *)
 GOTOXY(0,0);
 WRITELN('SELECT FROM THE FOLLOWING:
 WRITELN('    ');
 WRITELN('   0) RETURN TO MAIN MENU');
 WRITELN('   1) METALS');
 WRITELN('   2) LOW-MELTING METALS');
 WRITELN('   3) DUCTILE METALS');
 WRITELN('   4) BRITTLE METALS');
 WRITELN('   5) LIGHT METALS');
 WRITELN('   6) NON-METALS');
 WRITELN('   7) RARE GASES');
 WRITELN('   8) LANTHANIDE SERIES');
 WRITELN('   9) ACTINIDE SERIES');
 WRITELN('  10) TRANSITION ELEMENTS');
 WRITELN('  11) HALOGENS');
 WRITELN('    ');

 (* GET ANSWER AND IMPLEMENT NEEDED PROCEDURES *)
REPEAT
 GOTOXY(1,15);
 WRITELN('WHAT IS YOUR CHOICE
 GOTOXY(21,15);
 READ(C);
CASE C OF
  0:EXIT(GROUP);
  1:BEGIN P:='METALS';METALS;DUCT;RMET;END;
  2:BEGIN P:='LOW MELTING METALS';ES:=LO;WORK;END;
  3:BEGIN P:='DUCTILE METALS';DUCT;END;
  4:BEGIN P:='BRITTLE METALS';ES:=BR;WORK;END;
```

```
      5:BEGIN P:='LIGHT METALS';ES:=LM;WORK;END;
      6:BEGIN P:='NON-METALS';ES:=NM;WORK;END;
      7:BEGIN P:='RARE GASES';ES:=RG;WORK;END;
      8:BEGIN P:='LANTHANIDE';ES:=LA;WORK;END;
      9:BEGIN P:='ACTINIDE';ES:=AC;WORK;END;
      10:BEGIN P:='TRANSITION';ES:=TR;WORK;END;
      11:BEGIN P:='HALOGENS';ES:=HL;WORK;END
     END;(* OF CASE *);

     TEXTMODE;

   UNTIL (C<0) OR (C>11);

  END;

PROCEDURE DESCRIBE;
(*********************************)
(* THIS PROCEDURE PROVIDES THE   *)
(* USER WITH A SHORT DESCRIPTION *)
(* OF THE VARIOUS CHOICES        *)
(*********************************)

  BEGIN
   PAGE(OUTPUT);(* CLEARS TEXT SCREEN *)
   WRITELN('1) TEMP: ');
   WRITELN(' ');
   WRITELN('   A VISUAL DEMONSTRATION SHOWING THE ');
   WRITELN('MELTING AND BOILING POINTS OF VARIOUS ');
   WRITELN('ELEMENTS. TEMPERATURES WILL RANGE FROM ');
   WRITELN('-300 TO 6000 DEGREES CELSIUS ');
   WRITELN(' ');
   WRITELN(' ');
   WRITELN('2) STATUS: ');
   WRITELN(' ');
   WRITELN('   THE SAME AS TEMP BUT ALLOWS A ');
   WRITELN('TEMPERATURE TO BE CHOSEN');
   WRITELN(' ');
   WRITELN(' ');
   WRITELN('3) GROUP:');
   WRITELN(' ');
   WRITELN('   A VISUAL DESCRIPTION OF THE LOCATIONS');
   WRITELN('OF VARIOUS GROUPS OF ELEMENTS ON THE');
   WRITELN('PERIODIC TABLE');

   GOTOXY(5,22);
   WRITELN('HIT ANY KEY TO CONTINUE');
   GOTOXY(41,22);
  REPEAT
  UNTIL KEYPRESS;
   UNITCLEAR(1);(* CLEARS KEYPRESS *)
 END;

PROCEDURE CHOOSE;
(*************************************)
(*  THIS PROCEDURE IS WHERE MAIN     *)
(*  CHOICES ARE MADE AND APPROPRIATE *)
(*  PROCEDURES CALLED                *)
(*************************************)

  BEGIN
    PAGE(OUTPUT);(* CLEARS TEXT SCREEN *)
    TEXTMODE;

    (* PRINT CHOICES *)
    WRITELN('THIS PROGRAM CAN:');
    WRITELN(' ');
    WRITELN('   0) EXIT');
    WRITELN('   1) DESCRIBE CHOICES');
    WRITELN('   2) TEMPERATURE');
    WRITELN('   3) STATUS');
    WRITELN('   4) GROUPS');
    WRITELN('   ');
    WRITELN('   ');
    WRITELN('WHICH WOULD YOU PREFER?');
    GOTOXY(24,9);

   (* GET ANSWER AND IMPLEMENT NEEDED PROCEDURES THEN RETURN *)
   READ(C);
   CASE C OF
    1:BEGIN DESCRIBE;CHOOSE;END;
    2:BEGIN TEMP;CHOOSE;END;
    3:BEGIN STATUS;CHOOSE;END;
    4:BEGIN GROUP;CHOOSE;END;
   END;(* OF CASE *)
 END;

PROCEDURE INIT;
(*************************************)
(*  THIS PROCEDURE LOADS DISK FILE   *)
(*  NAMED 'PD2' INTO ACTIVE MEMORY   *)
(*************************************)

 VAR STORE: FILE OF DATA;

  BEGIN
   RESET(STORE,'PD2');
   I:=0;
   WHILE NOT EOF(STORE) DO
     BEGIN
       I:=I+1;
       ELEMENT[I]:=STORE↑;
       GET(STORE);
     END;(* OF WHILE *)
   CLOSE(STORE,LOCK);
END;

BEGIN(*MAIN PROGRAM*)
  INIT;
  CHOOSE;

END.
```

REMOVED FROM THE
ALVERNO COLLEGE LIBRARY

ALVERNO COLLEGE LIBRARY
Computers in science and social studies
371.39445C738s
2 5050 00276423 7

159034
371.39445
C738s

Alverno College
Library Media Center
Milwaukee, Wisconsin
DEMCO

$14.95

This diverse collection of 49 computer simulations and applications is a valuable source of ideas to make learning more realistic and more fun.

Twenty social studies simulations let you manage a kingdom, trade furs, operate a power plant, run an oil company, fight the Civil War or travel the Oregon Trail. Ecology simulations let you explore various ecosystems from the point of view of an animal, fish, or tree. Physics becomes a game with simulations of a lunar lander, drag racer and artillery cannon. You'll also find programs in genetics, psychology, health, astronomy and chemistry. The 49 intriguing, user-tested applications in this book emphasize learning by doing, as they encourage the user to ask the important "what if" questions. In this book you will find:

The Game of Life • Ecology Simulations • Life Expectancy Quiz • Lunar Landing Simulation • Chemistry Gas Laws Program • Hammurabi, Dukedom and King • Oregon Trail and Fur Trader • Nuclear Power Plant Simulations • Genetic Calculations and Problem Solving • Electric Company and Oil Company Simulations...and much more! The editor, David H. Ahl, is founder and president of *Creative Computing.*

CREATIVE COMPUTING PRESS

ISBN #0-916688-44-5

W7-BBG-306